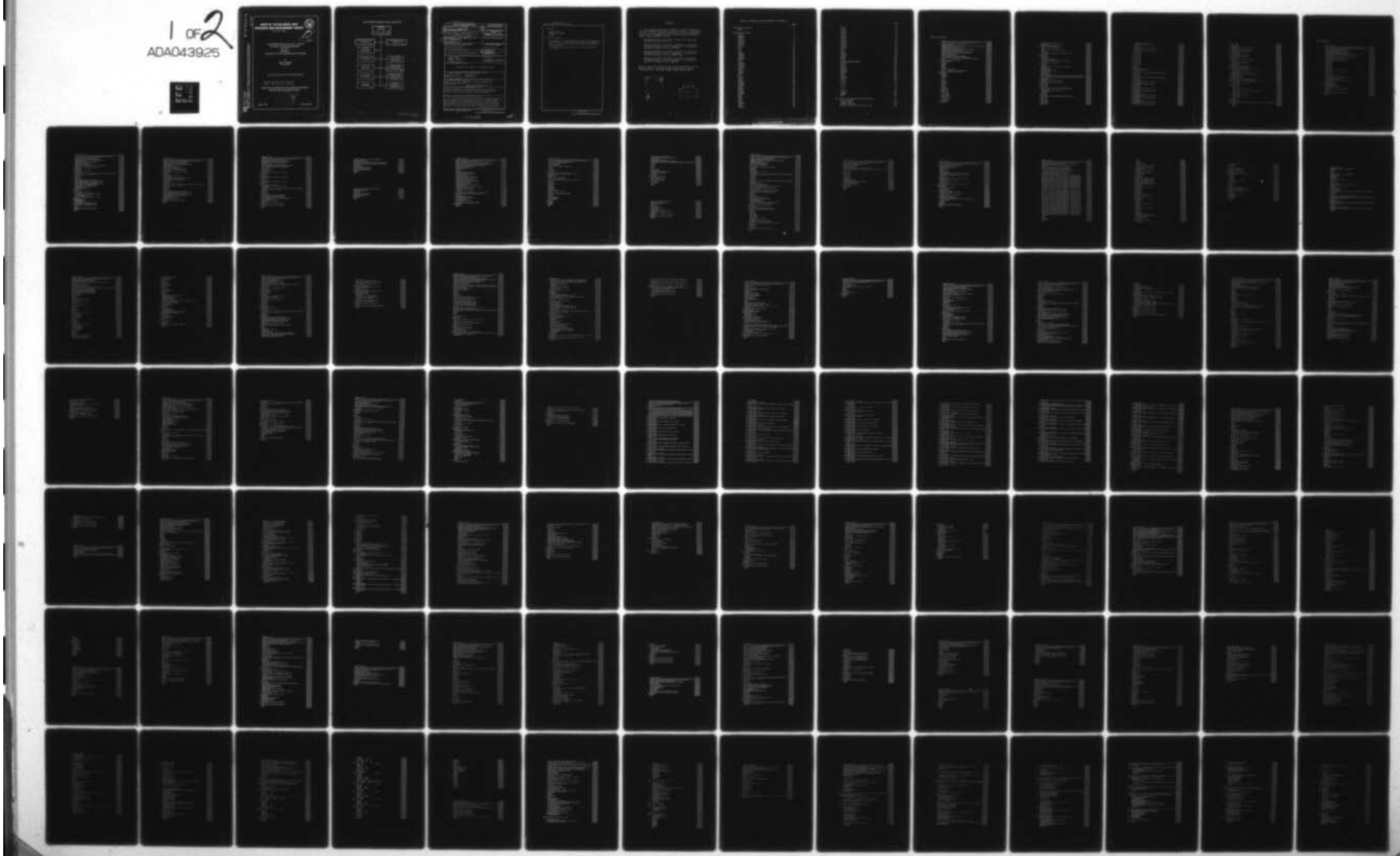
AD-A043 925   DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC   F/G 9/2
MAINTENANCE MANUAL FOR AUDIT. A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)
AUG 77   R J WYBRANIEC, R REGEN

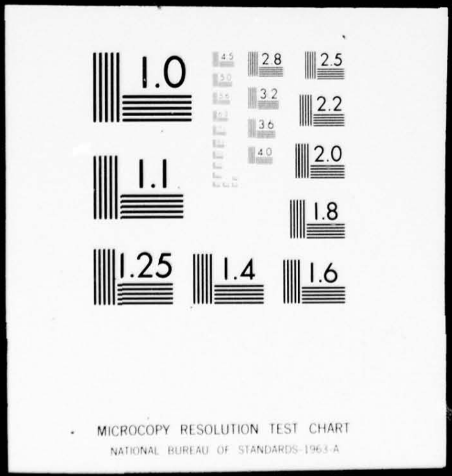
UNCLASSIFIED   DTNSRDC-77-0075-VOL-4   NL

1 OF 2
ADA043925



1 OF 2

ADA043925



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE
VOLUME 4 – APPENDIX D – LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360

Report 77-0075

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE

VOLUME 4

APPENDIX D

LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360

by

Robert J. Wybraniec
Richard Regen

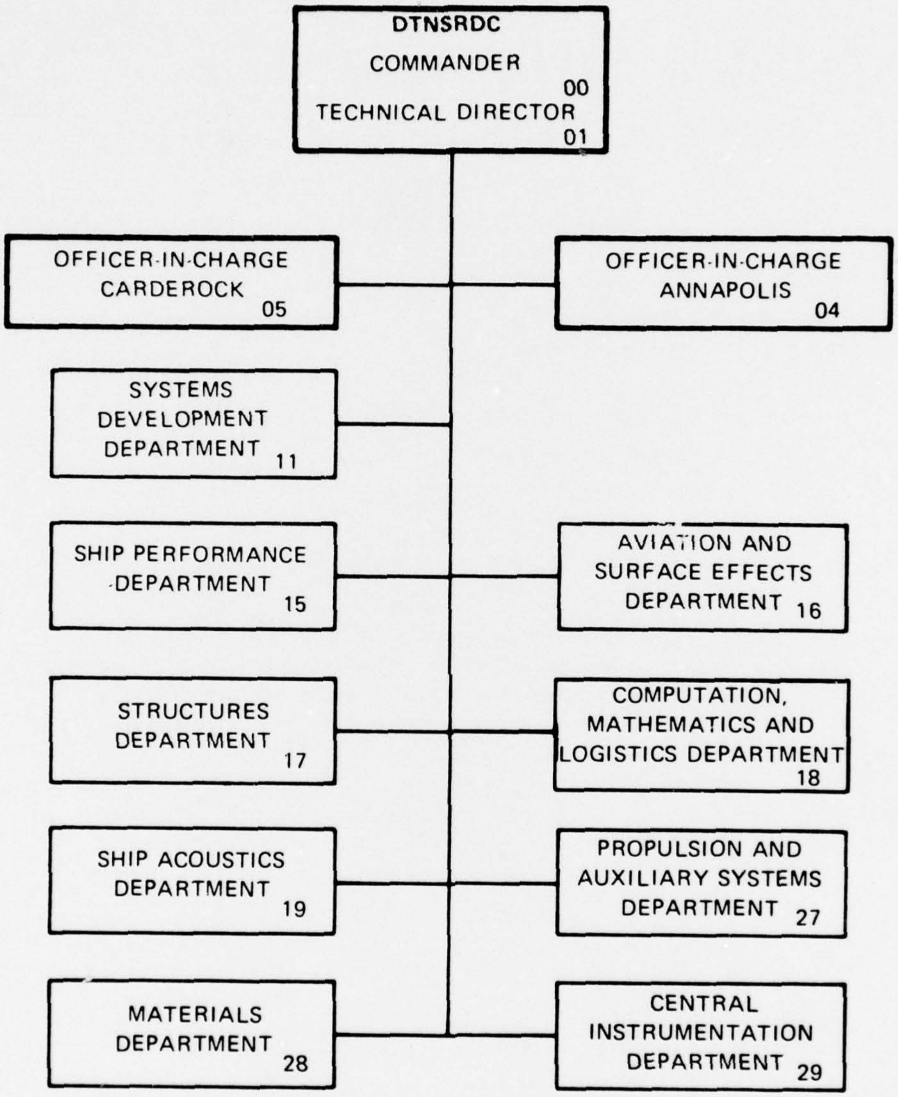
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

August 1977

Report 77-0075

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



DTNSRDC
COMMANDER 00
TECHNICAL DIRECTOR 01

OFFICER-IN-CHARGE CARDEROCK 05

OFFICER-IN-CHARGE ANNAPOLIS 04

SYSTEMS DEVELOPMENT DEPARTMENT 11

SHIP PERFORMANCE DEPARTMENT 15

AVIATION AND SURFACE EFFECTS DEPARTMENT 16

STRUCTURES DEPARTMENT 17

COMPUTATION, MATHEMATICS AND LOGISTICS DEPARTMENT 18

SHIP ACOUSTICS DEPARTMENT 19

PROPULSION AND AUXILIARY SYSTEMS DEPARTMENT 27

MATERIALS DEPARTMENT 28

CENTRAL INSTRUMENTATION DEPARTMENT 29

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

~~1. REPORT NUMBER~~
14 DTNSRDC-77-ØØ75-Vol-4

2. GOVT ACCESSION NO.
9

3. RECIPIENT'S CATALOG NUMBER

~~4. TITLE (and Subtitle)~~
6 MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE, VOLUME 4: APPENDIX D • LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360,

5. TYPE OF ~~REPORT & PERIOD CO~~VERED
Final Rept.

6. ~~P~~ERFORMING ORG. REPORT NUMBER

~~7. AUTHOR(s)~~
10 Robert J. Wybraniec
Richard Regen

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
David W. Taylor Naval Ship Research
and Development Center
Bethesda, Maryland 20084

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(See reverse side)

11. CONTROLLING OFFICE NAME AND ADDRESS
Navy Surface Effect Ships Project (PMS 304)
P.O. Box 34401 - Bethesda, Maryland 20084

~~12. REPORT DATE~~
11 August 1977

~~13. NUMBER OF PA~~GES
177

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
12 176p.

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT ~~(of the abstract entered in Block 20, if~~ different from Report)
16 SSH15, SØ3Ø8

18. S~~UPPLEMENTARY NOTES~~
17 SSH15ØØ1, SØ3Ø8ØØ1

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SESCOMP, SESCOMPSPEC's, Software Verification, Software Engineering, Reliability, Graph Theory, FORTRAN Software, Modules, Flow Analysis, Variable Precision Execution, Parser, Roll Call, Portability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The AUDIT documentation provides the maintenance programmer personnel with the information to effectively maintain and use the AUDIT software. The AUDIT software examines FORTRAN computer programs or modules developed under the SESCOMP system for compliance with certain prescribed standards (SESCOMPSPEC's) and produces reports detailing the deviations from those standards. The AUDIT software also examines a program unit to detect and

(Continued on reverse side)

→ next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387682

(Block 10)

63534N, 19588,
SSh15001 and S0308001,
11837001

(Block 20 continued)

report improper use of undefined variables along the program unit's possible paths. In addition, AUDIT has an option which enables the user to test the effect of changes in word length on the output of computer programs.

This report contains the listings of the AUDIT software for the IBM 360.

FOREWORD

The use and maintenance of AUDIT, a software system for analyzing SESCOMP contractor-supplied software, is documented as a set of four separately bound David W. Taylor Naval Ship Research and Development Center volumes sharing the common report number--DTNSRDC 77-0075:

- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 1

- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 2; Appendix B - Listings of the AUDIT Software for the CDC 6000

- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 3; Appendix C - Listings of the AUDIT Software for the UNIVAC 1108

- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 4; Appendix D - Listings of the AUDIT Software for the IBM 360

Volume 1 describes AUDIT and the use and maintenance of the AUDIT software. The other three volumes offer software listings for the CDC 6000, UNIVAC 1108, and IBM 360.

INDEX TO PROGRAMS AND SUBPROGRAMS IN APPENDIX D

Page

AUDIT Main Program

```
      COMMON A(1326).D(500).IDTBL(11.500).INITID(3).LASTID(3).ISRCH(3).
     * NXTID(2).JPTR,N,M,JTYP.LSTART.N2.IFNCNM,LOGID.IDTYP,NID.
     * LOC.LTYP.ITYP.IBLKDT.MODE,IERR,IDES
      DIMENSION IORD(15)
      COMMON/GLOBAL/NBLK.NREF.NSUBS.BLKTBL(200).EXTTBL(100).ISUBS(100)
      COMMON/INPOUT/NCALL.IN,IOP
      COMMON/LABELS/STATRA(2,200).NLABEL
      COMMON/LIST/NLIST.NINTFC,ISUBLT(4.200).INTFAC(600)
      COMMON/STFUNC/NSTFNC,ISTFNC(10)
      COMMON/DOLOOP/ISTACK(4.50).NSTACK.ILOOP.IOVFLW
      COMMON/BASBLK/IBLOCK(2500).NBLOCK.NB.NBRNCH
      COMMON/NTIMES/NTIMES.IQZ
      COMMON/FLOW/IFL.IRP
      INTEGER A.BLANK.STATRA.BLKTBL.EXTTBL
      DATA BLANK/1H /.IC/1HC/.IF/1HF/.MAIN/4HMAIN/
      DATA IORD/29.30.31.32.25.28.19.20.21.22.23.24.26.27.35/
      IOP=8
      READ(5.12) MODE.IN.IFL.INTR
   12 FORMAT(I1.3I4)
      NCALL=0
      NTIMES=0
      NREF=0
      NBLK=0
      NSUBS=0
      IRP=IFL-1
      READ(4) NLIST.NINTFC
      READ(4) ((ISUBLT(I.J).I=1.4).J=1.NLIST)
      READ(4) (INTFAC(I).I=1.NINTFC)
 1200 CONTINUE
    4 ISTAT=0
      IOVFLW=0
      MAINPR=0
      NLABEL=0
      NID=0
      JJ=0
      IBLKDT=0
      NSTACK=0
      NBLOCK=0
      ILOOP=0
      NB=0
      IBLKST=0
      NSTFNC=0
      WRITE(6.13)
   13 FORMAT(1H1)
      IFNCNM=0
      DO 2 I=1.11
      DO 2 J=1.500
    2 IDTBL(I.J)=0
      DO 3 I=1.200
      STATRA(1.I)=0
    3 STATRA(2.I)=0
      DO 5 I=1.3
      INITID(I)=0
    5 LASTID(I)=0
  700 CONTINUE
```

```
      IF(NBLOCK .GT. 2500) GO TO 7000
      CALL BUILD
      WRITE(6,1000) (A(I),I=1,N)
 1000 FORMAT(//6X,100A1,13(/6X,100A1))
      IF(A(1) .EQ. IC) GO TO 700
      IFST=1
      IF(NEXT(IFST) .EQ. BLANK) GO TO 700
      JJ=JJ+1
      CALL CLASS
      IF(ITYP .GT. 18 .AND. ITYP .NE. 28) GO TO 1320
      CALL STATNO
 1320 JPTR=7
      IPREC=ISTAT
      ISTAT=ITYP
      IF(JJ .EQ. 1) GO TO 7
      IF(IBLKDT .EQ. 0) GO TO 6
      IF(ITYP .GE. 18 .AND. ITYP .LE. 27) GO TO 8
      GO TO 220
    6 IF(LTYP .EQ. 9) GO TO 120
      GO TO 8
    7 IPREC=29
      IF(ITYP .GE. 29 .AND. ITYP .LE. 32) GO TO 25
      MAINPR=1
      NXTID(1)=MAIN
      NXTID(2)=BLANK
   25 IF(MAINPR .EQ. 0) GO TO 8
      CALL PROG
      WRITE(IOP,610)
      WRITE(IOP,620)
    8 GO TO (60,70,80,90,100,110,20,140,20,20,40,40,50,50,50,120,130,
     1 2000,10,10,10,10,10,170,180,190,200,210,150,38,30,160,220,220,
     2 220,220),ITYP
   10 DO 15 I=1,11
      IF(IPREC .EQ. IORD(I)) GO TO 17
   15 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
   17 CALL TYPE
      GO TO 500
   20 CALL SIMP
      GO TO 500
   30 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
      DO 35 I=1,20
      IF(NEXT(JPTR) .NE. IF) GO TO 35
      JPTR=JPTR-1
      CALL SUB
      GO TO 500
   35 CONTINUE
   38 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
      CALL SUB
      GO TO 500
   40 CALL IO
      GO TO 500
   50 CALL AUXIO
      GO TO 500
   60 CALL INIT
```

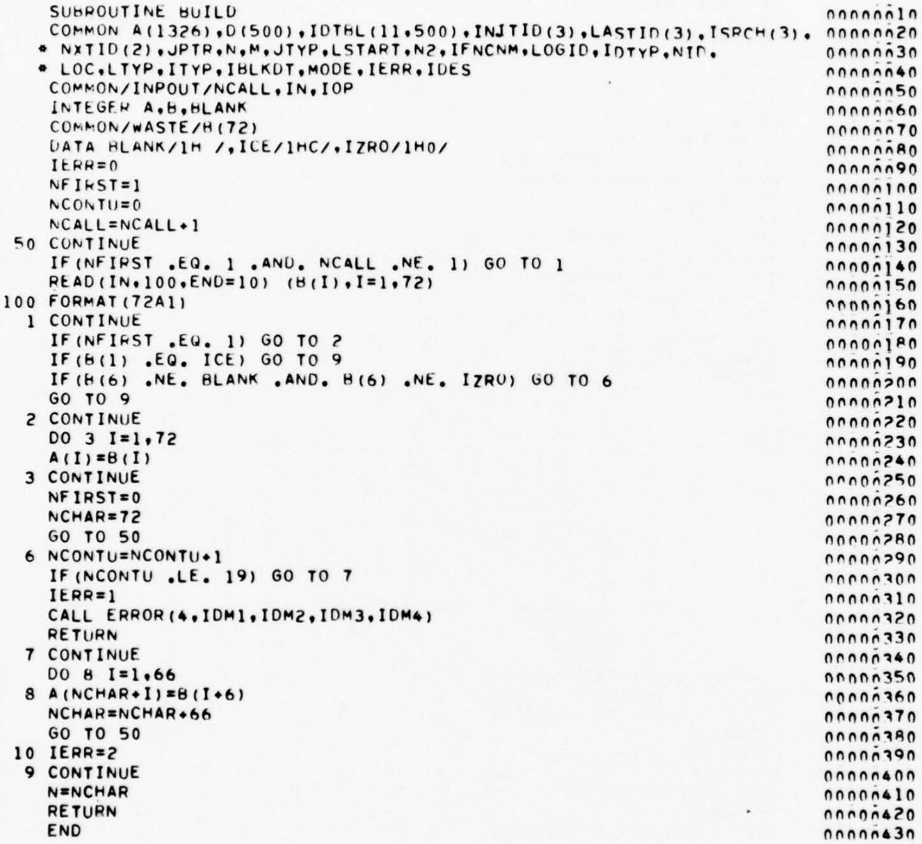
```
      WRITE(6,66) (A(I),I=1,N)
      IF(ITYP .NE. 35) GO TO 500
      ISTAT=35
      DO 67 I=1,15
      IF(IPREC .EQ. IORD(I)) GO TO 500
   67 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
      GO TO 500
   70 CALL ASSIGN
      GO TO 500
   80 CALL GOTO
      GO TO 500
   90 CALL ASGOTO
      GO TO 500
  100 CALL CTGOTO
      GO TO 500
  110 CALL ARIF
      GO TO 500
  120 CALL LOGIF
      GO TO 500
  130 CALL DO
      GO TO 500
  140 CALL CALL
      WRITE(6,66) (A(I),I=1,N)
   66 FORMAT(6X,72A1)
      GO TO 500
  150 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
      IBLKDT=1
      CALL SIMP
      GO TO 500
  160 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
      CALL PROG
      GO TO 500
  170 DO 175 I=1,12
      IF(IPREC .EQ. IORD(I)) GO TO 177
  175 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
  177 CALL DIMEN
      GO TO 500
  180 DO 185 I=1,5
      IF(IPREC .EQ. IORD(I)) GO TO 187
  185 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
  187 CALL COM
      GO TO 500
  190 DO 195 I=1,13
      IF(IPREC .EQ. IORD(I)) GO TO 197
  195 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
  197 CALL EQUIV
      GO TO 500
  200 DO 205 I=1,14
      IF(IPREC .EQ. IORD(I)) GO TO 207
  205 CONTINUE
      CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
```

```
 207 CALL DATA
     GO TO 500
 210 DO 215 I=1,6
     IF(IPREC .EQ. IORD(I)) GO TO 217
 215 CONTINUE
     CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)
 217 IF(N .GT. 72) GO TO 240
     WRITE(IOP,218) (A(I),I=1,N)
 218 FORMAT(72A1)
     GO TO 250
 240 WRITE(IOP,245) (A(I),I=1,N)
 245 FORMAT(72A1/(5X,1H*,66A1))
 250 CALL FRMAT
     GO TO 700
 220 CALL ERROR(1,IDM1,IDM2,IDM3,IDM4)
 500 CONTINUE
     IF(N .GT. 72) GO TO 540
     WRITE(IOP,520) (A(I),I=1,N)
 520 FORMAT(72A1)
     GO TO 600
 540 WRITE(IOP,545) (A(I),I=1,N)
 545 FORMAT(72A1/(5X,1H*,66A1))
 600 IF(MODE .NE. 1) GO TO 700
     IF(ITYP .LT. 30 .OR. ITYP .GT. 32) GO TO 700
     WRITE(IOP,610)
 610 FORMAT(5X,15H COMPLEX Q1COMP)
     WRITE(IOP,620)
 620 FORMAT(5X,24H DOUBLE PRECISION Q1DPRE)
     GO TO 700
2000 WRITE(IOP,2020)
2020 FORMAT(6X,3HEND)
     IF(N .NE. 72) WRITE(6,2100)
2100 FORMAT(6X,22H ILLEGAL END STATEMENT)
     IF(MAINPR .EQ. 1 .OR. IBLKDT .EQ. 1) GO TO 2200
     CALL SUBCHK
2200 IF(INTR .NE. 1) GO TO 3205
     CALL INTRIN
3205 CALL SYMTAB
     CALL GRT
     CALL COMCHK
     IF(IOVFLW .EQ. 1) GO TO 3210
     CALL LOOPCK
     IF(IFL .EQ. 0 .OR. IBLKDT .EQ. 1) GO TO 3210
     CALL FLOWCK
3210 IF(IERR .NE. 2) GO TO 4
     CALL GLOTAB
     IF(MODE .EQ. 1) GO TO 6000
     CALL GENROL
     REWIND 9
6000 REWIND 8
     STOP
7000 WRITE(6,7005)
7005 FORMAT(/////5X,54H OVERFLOW OF BASIC BLOCK TABLE - PROCESSING TERM
    *INATED)
     STOP
     END
```

AUDIT Subprograms

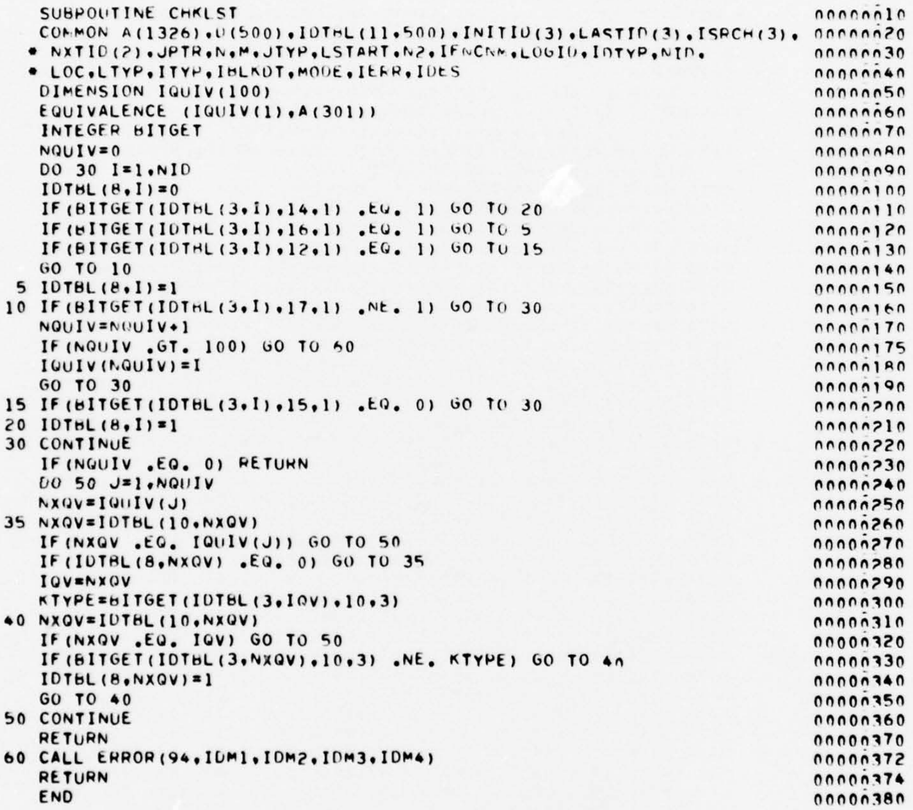
```
      SUBROUTINE ARIF
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,STATRA,STR,COMMA,BLANK,RHSTYP,AY,EF
      INTEGER BITPUT
      DATA LPAR/1H(/,COMMA/1H,/,BLANK/1H /,AY/1HI/,EF/1HF/
      IF(NEXT(JPTR) .NE. AY) GO TO 20
      IF(NEXT(JPTR) .NE. EF) GO TO 20
      IF(NEXT(JPTR) .NE. LPAR) GO TO 20
      JPTR=JPTR-1
      CALL EXPR
      NSTR=NSTR+1
      STR(NSTR)= -5
      NTYPE=1
      CALL PARSE
      CALL FNCSTR
      CALL BLKSTR
      IF(RHSTYP .EQ. 1) CALL ERROR(42)
      NBRNCH=0
      DO 10 I=1,3
      CALL GNLE
      IF(JTYP .NE. 5)'GO TO 20
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      IF( NBRNCH .EQ. 0) GO TO 5
      DO 3 J=1,NBRNCH
      IF(LOC .EQ. IBLOCK(NBLOCK-J+1)) GO TO 7
    3 CONTINUE
    5 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=LOC
      NBRNCH=NBRNCH+1
    7 IF(I .EQ. 3) GO TO 10
      IF(NEXT(JPTR) .NE. COMMA) GO TO 20
   10 CONTINUE
      IF(NEXT(JPTR) .NE. BLANK) GO TO 20
      NB=1
      RETURN
   20 CALL ERROR(7.IDM1.IDM2.IDM3.IDM4)
      RETURN
      END
      SUBROUTINE ASGOTO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(4)
      INTEGER STATRA,BLANK,COMMA,RPAR,A
      INTEGER BITPUT,BITGET
      DATA IALPH/1HG,1HO,1HT,1HO/
      DATA BLANK/1H /,COMMA/1H,/,LPAR/1H(/,RPAR/1H)/
      DO 5 I=1,4
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30
    5 CONTINUE
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 30
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 10
      IDTYP=1
      CALL STORE
      LOC=NID
   10 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(NEXT(JPTR) .NE. COMMA) GO TO 30
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=5000+LOC
      NBRNCH=0
   20 CALL GNLE
      IF(JTYP .NE. 5) GO TO 30
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      IF(NBRNCH .EQ. 0) GO TO 25
      DO 22 I=1,NBRNCH
      IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 27
   22 CONTINUE
   25 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=LOC
      NBRNCH=NBRNCH+1
   27 IF(NEXT(JPTR) .EQ. COMMA) GO TO 20
      IF(A(JPTR-1) .NE. RPAR) GO TO 30
      IF(NEXT(JPTR) .NE. BLANK) GO TO 30
      NB=1
      RETURN
   30 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE ASSIGN
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(6)
      INTEGER BLANK,TEE,OH,STATRA
      INTEGER BITPUT,BITGET
      DATA BLANK/1H /,TEE/1HT/,OH/1HO/
      DATA IALPH/1HA,1HS,1HS,1HI,1HG,1HN/
      DO 5 I=1,6
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 20
    5 CONTINUE
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 20
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      IF(NEXT(JPTR) .NE. TEE) GOTO 20
      IF(NEXT(JPTR) .NE. OH) GO TO 20
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 20
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 10
      IDTYP=1
      CALL STORE
      LOC=NID
   10 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(NEXT(JPTR) .NE. BLANK) GO TO 20
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=4000+LOC
      RETURN
   20 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE AUXIO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH1(6),IALPH2(9),IALPH3(7)
      INTEGER BITGET
      DATA IALPH1/1HR,1HE,1HW,1HI,1HN,1HD/
      DATA IALPH2/1HB,1HA,1HC,1HK,1HS,1HP,1HA,1HC,1HE/
      DATA IALPH3/1HE,1HN,1HD,1HF,1HI,1HL,1HE/
      DATA IBLANK/1H /
      IT=16-ITYP
      IF(IT-2) 25,15,5
    5 DO 10 I=1,6
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50
   10 CONTINUE
      GO TO 40
   15 DO 20 I=1,9
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50
   20 CONTINUE
      GO TO 40
   25 DO 30 I=1,7
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50
   30 CONTINUE
   40 CALL GNLE
      IF(JTYP .NE. 2) GO TO 60
      IF(NEXT(JPTR) .NE. IBLANK) GO TO 50
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 45
      IDTYP=1
      CALL STORE
      LOC=NID
   45 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)
      IF(BITGET(IDTHL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=2000+LOC
      RETURN
   50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
   60 CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      INTEGER FUNCTION BITGET(ILOC,IPOS,IWIDTH)
      DIMENSION IMASK(18)
      EQUIVALENCE (VAL,IVAL)
      DATA IMASK/Z1,Z3,Z7,ZF,Z1F,Z3F,Z7F,ZFF,Z1FF,Z3FF,
     * Z7FF,ZFFF,Z1FFF,Z3FFF,Z7FFF,ZFFFF,Z1FFFF,Z3FFFF/
      JLOC=ILOC
      NSHIFT=32-IPOS
      IF(NSHIFT .EQ. 0) GO TO 5
      CALL SHIFTR(JLOC,NSHIFT)
    5 VAL=AND(JLOC,IMASK(IWIDTH))
      BITGET=IVAL
      RETURN
      END
```

```
      INTEGER FUNCTION BITPUT(ILOC,IVAL,IPOS)
      EQUIVALENCE (RNEW,NEW)
      NSHIFT=32-IPOS
      JVAL=IVAL
      IF(NSHIFT .EQ. 0) GO TO 5
      CALL SHIFTL(JVAL,NSHIFT)
    5 RNEW=OR(ILOC,JVAL)
      BITPUT=NEW
      RETURN
      END
      SUBROUTINE BLKSTR
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER BITPUT,BITGET,FNCLOC
      IF(MARGS .EQ. 0) RETURN
      DO 100 I=1,MARGS
      IOSTAT=2
      ICOL=9
      LOC=BITGET(IARGS(I),ICOL,9)
      NFNC=BITGET(IARGS(I),ICOL+3,3)
      IF(NFNC .EQ. 0) GO TO 60
      ILOC=FNCLOC(NFNC)
      NARG=BITGET(IARGS(I),ICOL+9,6)
      INDEX=BITGET(IDTBL(3,ILOC),32,9)
      IF(INDEX .EQ. 0) GO TO 60
      IPTR=ISUBLT(4,INDEX)+(NARG-1)/3
      JVAR=BITGET(ISUBLT(3,INDEX ),14,1)
      ICOL=9*MOD(NARG-1,3)+8
      IF(JVAR .EQ. 1) ICOL=8
      IOSTAT=BITGET(INTFAC(IPTR),ICOL,2)
      KPTR=(NARG+5)/3
      IEXP=BITGET(IFNCRA(NFNC,KPTR),ICOL+1,1)
      IF(IOSTAT .EQ. 2) GO TO 60
      IF(IEXP .NE. 0) GO TO 40
      IF(IOSTAT .EQ. 1) GO TO 60
      GO TO 80
   40 IF(BITGET(ISUBLT(3,INDEX),10,4) .NE. 0) GO TO 90
      INTFAC(IPTR)=BITPUT(INTFAC(IPTR),2,ICOL)
      IOSTAT=2
   60 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=2000+LOC
      IF(IOSTAT .EQ. 2) GO TO 100
   80 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=1000+LOC
      GO TO 100
   90 CALL ERROR(55,NARG,IDM2,IDM3,IDM4)
  100 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE BUILD
      COMMON A(1326),D(500),IDTBL(11,500),INJTID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/INPOUT/NCALL,IN,IOP
      INTEGER A,B,BLANK
      COMMON/WASTE/B(72)
      DATA BLANK/1H /,ICE/1HC/,IZRO/1H0/
      IERR=0
      NFIRST=1
      NCONTU=0
      NCALL=NCALL+1
   50 CONTINUE
      IF(NFIRST .EQ. 1 .AND. NCALL .NE. 1) GO TO 1
      READ(IN,100,END=10) (B(I),I=1,72)
  100 FORMAT(72A1)
    1 CONTINUE
      IF(NFIRST .EQ. 1) GO TO 2
      IF(B(1) .EQ. ICE) GO TO 9
      IF(B(6) .NE. BLANK .AND. B(6) .NE. IZRO) GO TO 6
      GO TO 9
    2 CONTINUE
      DO 3 I=1,72
      A(I)=B(I)
    3 CONTINUE
      NFIRST=0
      NCHAR=72
      GO TO 50
    6 NCONTU=NCONTU+1
      IF(NCONTU .LE. 19) GO TO 7
      IERR=1
      CALL ERROR(4,IDM1,IDM2,IDM3,IDM4)
      RETURN
    7 CONTINUE
      DO 8 I=1,66
    8 A(NCHAR+I)=B(I+6)
      NCHAR=NCHAR+66
      GO TO 50
   10 IERR=2
    9 CONTINUE
      N=NCHAR
      RETURN
      END
      SUBROUTINE CAA(ISTR,MSTR,ID)
      DIMENSION ISTR(6),ID(2),RID(2),JD(2)
      REAL K,IBLANK
      EQUIVALENCE (JD(1),RID(1))
      DATA MASK/ZFF000000/,IBLANK/Z40000000/
      IF(MSTR .GT. 6 .AND. ITYP .NE. 28) CALL ERROR(6,IDM1,IDM2,IDM3,I4)
      JD(1)=0
      JD(2)=0
      J=0
      NCHAR=0
    5 J=J+1
      DO 20 I=1,4
      NCHAR=NCHAR+1
      IF(NCHAR .GT. MSTR) GO TO 10
      K=AND(ISTR(NCHAR),MASK)
      GO TO 15
   10 K=IBLANK
   15 IF(I .EQ. 1) GO TO 20
      NSHIFT=8*(I-1)
      CALL SHIFTR(K,NSHIFT)
   20 RID(J)=OR(RID(J),K)
      IF(J .EQ. 1) GO TO 5
      ID(1)=JD(1)
      ID(2)=JD(2)
      RETURN
      END
```

```
      SUBROUTINE CAI(ISTR,MSTR,INTVAL)
      DIMENSION ISTR(10)
      IF(MSTR .GT. 10) GO TO 20
      INTVAL=0
      DO 10 I=1,MSTR
      INT=ISTR(I)
      CALL SHIFTR(INT,24)
      INT=INT-240
      IF(INT .EQ. 0) GO TO 10
      INTVAL=INTVAL+INT*10**(MSTR-I)
   10 CONTINUE
      IF(INTVAL .GT. 2**31-1) GO TO 20
      RETURN
   20 CALL ERROR(3,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE CALL
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      DIMENSION IALPH(4)
      INTEGER BLANK,BITPUT,BITGET
      DATA IALPH/1HC,1HA,1HL,1HL/
      DATA LPAR/1H(/,BLANK/1H /
      DO 5 I=1,4
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 50
    5 CONTINUE
      IPTR=JPTR
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 50
      CALL SEARCH
      IF(ISRCH(1) .EQ. 1) CALL ERROR(24,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(2) .EQ. 1) GO TO 8
      IDTYP=2
      CALL STORE
      LOC=NID
    8 CONTINUE
      IF(LOC .EQ. IFNCNM) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      ILOC=LOC
      NXT=NEXT(JPTR)
      IF(NXT .EQ. LPAR) GO TO 17
      IF(NXT .NE. BLANK) GO TO 50
      IF(BITGET(IDTBL(3,LOC),18,1) .EQ. 1) GO TO 20
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)
      DO 10 I=1;NLIST
      IF(IDTBL(1,LOC) .NE. ISUBLT(1,I)) GO TO 10
      IF(IDTBL(2,LOC) .NE. ISUBLT(2,I)) GO TO 10
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,32)
      LISTLC=I
      GO TO 22
   10 CONTINUE
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,LOC)
      ISUBLT(2,NLIST)=IDTBL(2,LOC)
      ISUBLT(3,NLIST)=0
      ISUBLT(4,NLIST)=0
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,32)
      CALL ERROR(52,IDM1,IDM2,IDM3,IDM4)
      RETURN
   20 LISTLC=BITGET(IDTBL(3,LOC),32,9)
   22 CONTINUE
      IF(BITGET(ISUBLT(3,LISTLC),6,6) .NE. 0)
     $ CALL ERROR(26,IDM1,IDM2,IDM3,IDM4)
      RETURN
   17 JPTR=IPTR
      NTYPE=1
      CALL EXPR
      CALL PARSE
      CALL FNCSTR
      CALL BLKSTR
      IF(MODE .EQ. 1) GO TO 48
      LOC=ILOC
      INDEX=BITGET(IDTBL(3,LOC),32,9)
      KLAS=BITGET(ISUBLT(3,INDEX),10,4)
      IF(KLAS .EQ. 1 .OR. KLAS .EQ. 2) CALL CALL2
      RETURN
   48 JPTR=IPTR-1
      CALL CNVRT
      RETURN
   50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE CALL2
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER A,D,COMMA,RPAR
      INTEGER BITPUT,BITGET
      DIMENSION IALPH(13)
      DATA RPAR/1H)/,COMMA/1H,/,I1/1H1/,IH/1HH/
      DATA IALPH/1HC,1HA,1HL,1HL,1H ,1HR,1HO,1HL,1HC,1HH,1HK,1H ,1H(/
      DO 15 J=1,13
      K=J+6
      A(K)=IALPH(J)
   15 CONTINUE
      DO 20 I=1,6
      KK=19+4*I
      A(KK-3)=I1
      A(KK-2)=IH
      IWRD=1+(I-1)/4
      IPOS=8*I-32*(IWRD-1)
      IVL=BITGET(IDTBL(IWRD,LOC),IPOS,8)
      A(KK-1)=BITPUT(0,IVL,8)
      IF(I .EQ. 6) GO TO 25
   20 A(KK)=COMMA
   25 A(KK)=RPAR
      N=KK
      RETURN
      END
```



```
      SUBROUTINE CHKLST
      COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IQUIV(100)
      EQUIVALENCE (IQUIV(1),A(301))
      INTEGER BITGET
      NQUIV=0
      DO 30 I=1,NID
      IDTBL(8,I)=0
      IF(BITGET(IDTBL(3,I),14,1) .EQ. 1) GO TO 20
      IF(BITGET(IDTBL(3,I),16,1) .EQ. 1) GO TO 5
      IF(BITGET(IDTBL(3,I),12,1) .EQ. 1) GO TO 15
      GO TO 10
    5 IDTBL(8,I)=1
   10 IF(BITGET(IDTBL(3,I),17,1) .NE. 1) GO TO 30
      NQUIV=NQUIV+1
      IF(NQUIV .GT. 100) GO TO 60
      IQUIV(NQUIV)=I
      GO TO 30
   15 IF(BITGET(IDTBL(3,I),15,1) .EQ. 0) GO TO 30
   20 IDTBL(8,I)=1
   30 CONTINUE
      IF(NQUIV .EQ. 0) RETURN
      DO 50 J=1,NQUIV
      NXQV=IQUIV(J)
   35 NXQV=IDTBL(10,NXQV)
      IF(NXQV .EQ. IQUIV(J)) GO TO 50
      IF(IDTBL(8,NXQV) .EQ. 0) GO TO 35
      IQV=NXQV
      KTYPE=BITGET(IDTBL(3,IQV),10,3)
   40 NXQV=IDTBL(10,NXQV)
      IF(NXQV .EQ. IQV) GO TO 50
      IF(BITGET(IDTBL(3,NXQV),10,3) .NE. KTYPE) GO TO 40
      IDTBL(8,NXQV)=1
      GO TO 40
   50 CONTINUE
      RETURN
   60 CALL ERROR(94,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
```

```
 SUBROUTINE CLASS
 COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
 DIMENSION KALP(48),KSUC(48),KFAL(48),KDEC(10),KF(8)
 INTEGER A
 DATA KDEC(1),KDEC(2),KDEC(3),KDEC(4),KDEC(5),
1  KDEC(6),KDEC(7),KDEC(8),KDEC(9),KDEC(10)
2  /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
 DATA KF(1),KF(2),KF(3),KF(4),KF(5),KF(6),KF(7),KF(8)
1   /1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN/
 DATA KBLNK,KLPAR,KRPAR,KEQ/1H ,1H(,1H),1H=/
 DATA KH,KSLSH,KASTK,KCMA/1HH,1H/,1H*,1H,/
 DATA KALP( 1),KALP( 2),KALP( 3),KALP( 4) /1HI,1HF,1HN,1HG/
 DATA KALP( 5),KALP( 6),KALP( 7),KALP( 8) /1HO,1HT,1HO,1H(/
 DATA KALP( 9),KALP(10),KALP(11),KALP(12) /1HC,1HA,1HO,1HN/
 DATA KALP(13),KALP(14),KALP(15),KALP(16) /1HM,1HM,1HP,1HR/
 DATA KALP(17),KALP(18),KALP(19),KALP(20) /1HE,1HA,1HD,1HL/
 DATA KALP(21),KALP(22),KALP(23),KALP(24) /1HT,1HW,1HF,1HO/
 DATA KALP(25),KALP(26),KALP(27),KALP(28) /1HU,1HD,1HI,1HA/
 DATA KALP(29),KALP(30),KALP(31),KALP(32) /1HO,1HU,1HW,1HS/
 DATA KALP(33),KALP(34),KALP(35),KALP(36) /1HT,1HU,1HE,1HN/
 DATA KALP(37),KALP(38),KALP(39),KALP(40) /1HD,1HF,1HX,1HQ/
 DATA KALP(41),KALP(42),KALP(43),KALP(44) /1HB,1HA,1HL,1HA/
 DATA KALP(45),KALP(46),KALP(47),KALP(48)/1HL,1HP,1HR,1HO/
 DATA KSUC( 1),KSUC( 2),KSUC( 3),KSUC( 4) /  2, -6,-19,  5/
 DATA KSUC( 5),KSUC( 6),KSUC( 7),KSUC( 8) /  6,  7,  8, -5/
 DATA KSUC( 9),KSUC(10),KSUC(11),KSUC(12) / 10, -9, 12, -7/
 DATA KSUC(13),KSUC(14),KSUC(15),KSUC(16) / 14,-25,-22, 17/
 DATA KSUC(17),KSUC(18),KSUC(19),KSUC(20) / 18, 19,-11,-20/
 DATA KSUC(21),KSUC(22),KSUC(23),KSUC(24) / -9,-13, 24,-29/
 DATA KSUC(25),KSUC(26),KSUC(27),KSUC(28) /-31, 27,-24,-27/
 DATA KSUC(29),KSUC(30),KSUC(31),KSUC(32) / 30,-21,-12, 33/
 DATA KSUC(33),KSUC(34),KSUC(35),KSUC(36) /-10,-30, 36, 37/
 DATA KSUC(37),KSUC(38),KSUC(39),KSUC(40) / 38,-15,-34,-26/
 DATA KSUC(41),KSUC(42),KSUC(43),KSUC(44) / 42,-14,-29, -2/
 DATA KSUC(45),KSUC(46),KSUC(47),KSUC(48) /-23,47,48,-32/
 DATA KFAL( 1),KFAL( 2),KFAL( 3),KFAL( 4) /  4,  3,-36,  9/
 DATA KFAL( 5),KFAL( 6),KFAL( 7),KFAL( 8) /-36,-36,-36, -3/
 DATA KFAL( 9),KFAL(10),KFAL(11),KFAL(12) / 16, 11,-36, 13/
 DATA KFAL(13),KFAL(14),KFAL(15),KFAL(16) /-36, 15,-36, 23/
 DATA KFAL(17),KFAL(18),KFAL(19),KFAL(20) /-36, 21, 20,-36/
 DATA KFAL(21),KFAL(22),KFAL(23),KFAL(24) / 22,-36, 26, 25/
 DATA KFAL(25),KFAL(26),KFAL(27),KFAL(28) /-36, 31, 28, 29/
 DATA KFAL(29),KFAL(30),KFAL(31),KFAL(32) /-36,-36, 32, 35/
 DATA KFAL(33),KFAL(34),KFAL(35),KFAL(36) / 34,-36, 41, 39/
 DATA KFAL(37),KFAL(38),KFAL(39),KFAL(40) /-36,-18, 40,-36/
 DATA KFAL(41),KFAL(42),KFAL(43),KFAL(44) / 44, 43,-36, 45/
 DATA KFAL(45),KFAL(46),KFAL(47),KFAL(48)/ 46,-36,-36,-36/
 LTYP=0
 IPTR=7
5 CONTINUE
 JSAVE=KBLNK
 JSW=0
 ISW=0
```

```
      JEQ=0
      JCMA=0
      JHOLL=0
      DO 26 J=IPTR,N
      JCH=A(J)
      IF(JCH .EQ. KBLNK) GO TO 26
    6 IF(JHOLL .LE. 0) GO TO 12
    7 DO 8 L=1,10
      IF(JCH .EQ. KDEC(L)) GO TO 10
    8 CONTINUE
      IF(JHOLL .LE. 1) GO TO 11
    9 IF(JCH .EQ. KH) GO TO 32
      GO TO 11
   10 JHOLL=JHOLL+1
      GO TO 25
   11 JHOLL=0
   12 IF(JCH .EQ. KLPAR) GO TO 20
   13 IF(JCH .EQ. KRPAR) GO TO 18
   14 IF(JCH .EQ. KCMA) GO TO 22
   15 IF(JCH .EQ. KEQ) GO TO 23
   16 IF(JCH .EQ. KSLSH) GO TO 21
   17 IF(JCH .EQ. KASTK) GO TO 21
      GO TO 25
   18 JSW=JSW-1
      IF(JSW .GT. 0) GO TO 25
   19 ISW=1
      GO TO 26
   20 JSW=JSW+1
   21 JHOLL=1
      GO TO 25
   22 IF(JSW) 30,30,21
   23 IF(JSW .GT. 0) GO TO 32
   24 JEQ=1
   25 IF(ISW .GT. 0) GO TO 27
   26 CONTINUE
      GO TO 28
   27 JSAVE=JCH
      JP=J
   28 IF(JEQ .LE. 0) GO TO 32
   29 JTP=1
      GO TO 55
   30 JCMA=1
      IF(JEQ .LE. 0) GO TO 32
   31 JTP=17
      GO TO 55
   32 J=1
      ISW=IPTR
   33 JCH=A(ISW)
      IF(JCH .EQ. KBLNK) GO TO 37
   34 IF(JCH .EQ. KALP(J)) GO TO 36
   35 J=KFAL(J)
      IF (J) 39,39,34
   36 J=KSUC(J)
      IF(J .LE. 0) GO TO 39
   37 ISW=ISW+1
```

```
      IF(ISW .LE. N) GO TO 33
   38 JCH=KBLNK
      GO TO 35
   39 JTP=-J
      IF(JTP-3) 55.45.40
   40 IF(JTP-6) 55.43.41
   41 IF(JTP .LT. 19) GO TO 55
   42 IF(JTP - 23) 47.47,55
   43 DO 44 L=1.10
      IF(JSAVE .EQ. KDEC(L)) GO TO 55
   44 CONTINUE
      LTYP=9
      JTP=16
      IPTR=JP
      GO TO 5
   45 IF(JCMA .LE. 0) GO TO 55
      JTP=4
      GO TO 55
   47 L=11
      GO TO 52
   48 L=L+1
      IF(L .GT. N) GO TO 55
   49 IF(A(L) .EQ. KBLNK) GO TO 48
   50 IF(A(L) .EQ. KF(ISW)) GO TO 53
   51 IF(ISW .EQ. 1) GO TO 48
   52 ISW=1
      GO TO 50
   53 ISW=ISW+1
      IF(ISW .LE. 8) GO TO 48
   54 JTP=31
   55 ITYP=JTP
      RETURN
      END
      SUBROUTINE CMPARE
      DIMENSION IROLCK(2,100),ISUB(2,100)
      REWIND 3
      NSUB=0
      NC=0
      DO 5 I=1,100
      READ(9,3,END=7) ISUB(1,I),ISUB(2,I)
    3 FORMAT(A4,A2)
      NSUB=NSUB+1
    5 CONTINUE
    7 IF(NSUB .EQ. 0) RETURN
      DO 40 L=1,12
      MODE=L-1
      NROLL=0
      DO 10 I=1,100
      READ(3,3,END=15) IROLCK(1,I),IROLCK(2,I)
      NROLL=NROLL+1
   10 CONTINUE
   15 IF(NROLL .EQ. 0) GO TO 35
      DO 30 J=1,NSUB
      DO 20 K=1,NROLL
      IF(IROLCK(1,K) .EQ. ISUB(1,J) .AND. IROLCK(2,K) .EQ. ISUB(2,J))
     * GO TO 30
   20 CONTINUE
      NC=1
      WRITE(6,25) ISUB(1,J),ISUB(2,J),MODE
   25 FORMAT(6X,12H SUBROUTINE ,A4,A2,53H WAS NOT CALLED IN THE ROLL CAL
     *L MODE FOR MODE INDEX ,I3)
   30 CONTINUE
      GO TO 40
   35 WRITE(6,36) MODE
   36 FORMAT(6X,65H NO SUBROUTINES WERE CALLED IN THE ROLL CALL MODE FOR
     * MODE INDEX ,I3)
      NC=1
   40 CONTINUE
      IF(NC .EQ. 0) WRITE(6,50)
   50 FORMAT(6X,50H ALL SUBROUTINES WERE CALLED IN THE ROLL CALL MODE)
      RETURN
      END
      SUBROUTINE CNVRT
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/STRING/NTYPE,NSTR,STR(500)
      DIMENSION IOP(8),ILOG(2,7),JLOG(3,2),IFUNC1(6),IFUNC2(6),
     1 IFUNC3(6)
      INTEGER STR,A,D,EQUALS,DECPT
      INTEGER BITPUT,BITGET
      DATA IOP/1H+,1H-,1H/,1H(,1H),1H,,1H*,1H*/
      DATA ILOG/1HL,1HT,1HL,1HE,1HG,1HT,1HG,1HE,1HE,1HQ,1HN,1HE,1HO,1HR/
      DATA JLOG/1HA,1HN,1HD,1HN,1HO,1HT/
      DATA IFUNC1/1HQ,1H1,1HR,1HE,1HA,1HL/
      DATA IFUNC2/1HQ,1H1,1HD,1HP,1HR,1HE/
      DATA IFUNC3/1HQ,1H1,1HC,1HO,1HM,1HP/
      DATA DECPT/1H./
      DO 5 J=1,JPTR
    5 D(J)=A(J)
      J=JPTR
      DO 100 K=1,NSTR
      IF(STR(K) .GT. 0) GO TO 40
      DO 10 I=1,8
      IF(STR(K) .NE. -I) GO TO 10
      D(J+1)=IOP(I)
      N3=1
      IF(I .NE. 8) GO TO 100
      D(J+2)=IOP(I)
      N3=2
      GO TO 100
   10 CONTINUE
      DO 15 I=1,7
      L=I+8
      IF(STR(K) .NE. -L) GO TO 15
      D(J+1)=DECPT
      D(J+2)=ILOG(1,I)
      D(J+3)=ILOG(2,I)
      D(J+4)=DECPT
      N3=4
      GO TO 100
   15 CONTINUE
      DO 20 I=1,2
      L=I+15
      IF(STR(K) .NE. -L) GO TO 20
      D(J+1)=DECPT
      D(J+2)=JLOG(1,I)
      D(J+3)=JLOG(2,I)
      D(J+4)=JLOG(3,I)
      D(J+5)=DECPT
      N3=5
      GO TO 100
   20 CONTINUE
      KL=1
      IF(STR(K) .EQ. -0) KL=2
      IF(STR(K) .EQ. -10000) KL=4
      IF(STR(K) .EQ. -20000) KL=3
      GO TO(110,25,30,35),KL
   25 DO 27 I=1,6
      D(J+I)=IFUNC1(I)
   27 CONTINUE
      N3=6
      GO TO 100
   30 DO 32 I=1,6
      D(J+I)=IFUNC2(I)
   32 CONTINUE
      N3=6
      GO TO 100
   35 DO 37 I=1,6
      D(J+I)=IFUNC3(I)
   37 CONTINUE
      N3=6
      GO TO 100
   40 IF(STR(K) .LT. 1000001) GO TO 110
      N3=STR(K)/1.E6
      NLOC=(STR(K)/10000)*10000
      JPTR=STR(K)-NLOC
      KLOC=STR(K)-N3*1.E6
      IF(KLOC .LT. 400000 .OR. KLOC .GT. 500000) GO TO 50
      IHL=(KLOC-400000)/10000
      IF(IHL .EQ. 5) GO TO 60
      DO 45 I=1,N3
      D(J+I)=NEXT(JPTR)
   45 CONTINUE
      GO TO 100
   50 DO 55 I=1,N3
      IWRD=(I-1)/4+1
      IPOS=8*I-32*(IWRD-1)
      ICHAR=BITGET(IDTBL(IWRD,JPTR),IPOS,8)
   55 D(J+I)=BITPUT(0,ICHAR,8)
      GO TO 100
   60 KPTR=JPTR-1
      DO 65 I=1,N3
   65 D(J+I)=A(KPTR+I)
  100 J=J+N3
      N=J
      DO 105 I=1,N
  105 A(I)=D(I)
      RETURN
  110 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE COM
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKOT,MODE,IERR,IDES
      DIMENSION IDIM(3),IALPH(6)
      INTEGER SLASH,COMMA,BLANK,A,RPAR
      INTEGER BITPUT,BITGET
      DATA IALPH/1HC,1HO,1HM,1HM,1HO,1HN/
      DATA SLASH/1H//,COMMA/1H,/,BLANK/1H /,RPAR/1H)/,LPAR/1H(/
      DO 10 I=1,6
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60
   10 CONTINUE
      IF(NEXT(JPTR) .EQ. SLASH) GO TO 15
      JPTR=JPTR-1
   12 NXTID(1)=BLANK
      NXTID(2)=BLANK
      GO TO 20
   15 CALL GNLE
      IF(A(JPTR-1) .EQ. SLASH) GO TO 12
      IF(JTYP .NE. 2) GO TO 60
      IF(NEXT(JPTR) .NE. SLASH) GO TO 60
   20 CALL COMSCH
      IF(ISRCH(3) .EQ. 1) GO TO 25
      IDTYP=3
      CALL STORE
      ICMLOC=NID
      GO TO 27
   25 ICMLOC=LOC
      LSTLOC=IDTBL(9,LOC)
   27 CALL GNLE
      IF(JTYP .NE. 2) GO TO 60
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 28
      IDTYP=1
      CALL STORE
      LOC=NID
   28 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)
     $ CALL ERROR(17,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1)
     $ CALL ERROR(53,NXTID(1),NXTID(2),IDM3,IDM4)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,16)
      ICMSIZ=1
      IF(NEXT(JPTR) .NE. LPAR) GO TO 40
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 80
      I=0
   35 I=I+1
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 60
      IDIM(I)=N2
      ICMSIZ=ICMSI7*N2
      IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 35
      IF(A(JPTR-1) .NE. RPAR) GO TO 60
      K=NEXT(JPTR)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)
      IF(I .GT. 3) GO TO 60
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)
      DO 30 J=1,I
   30 IDTBL(4+J,LOC)=IDIM(J)
   40 IF(IDTBL(8,ICMLOC) .EQ. 0) GO TO 45
      IDTBL(8,LSTLOC)=LOC
      GO TO 47
   45 IDTBL(8,ICMLOC)=LOC
   47 IDTBL(5,ICMLOC)=IDTBL(5,ICMLOC)+ICMSIZ
      IDTBL(9,ICMLOC)=LOC
      IDTBL(9,LOC)=ICMLOC
      LSTLOC=LOC
      IF(A(JPTR-1) .EQ. COMMA) GO TO 27
      IF(A(JPTR-1) .NE. SLASH) GO TO 50
      IDTBL(8,LOC)=IDTBL(8,ICMLOC)
      GO TO 15
   50 IF(NEXT(JPTR) .NE. BLANK) GO TO 60
      IDTBL(8,LOC)=IDTBL(8,ICMLOC)
      RETURN
   60 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
   80 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
      END
      SUBROUTINE COMCHK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      INTEGER BITGET,CMBLK(2,20),TP,SZ,PREVTP,BLKTBL
      INTEGER SESCOM(2,13),SESERR,SES(2),BLANK
      DIMENSION ITPS(6),IORD(6)
      DATA IORD/1,2,5,4,3,6/
      DATA SESCOM/4HCASE,1H ,3HINA,1H ,3HINB,1H ,3HINC,1H ,3HIOX,1H ,
     $ 4HNPAG,1HX,4HLINX,1H ,3HIOY,1H ,4HNPAG,1HY,4HLINY,1H ,3HIOZ,1H ,
     $ 4HNPAG,1HZ,4HLINZ,1H /
      DATA BLANK/1H /,IBKCOM/2H///
      DATA SES/4HSESC,2HOM/
      SESERR=0
      NSES=0
      ICTGR2=0
      IBLK=INITID(3)
      MODCLS=0
      LC2=BITGET(IDTBL(3,1),32,9)
      IF(LC2 .EQ. 0 .OR. IBLKDT .EQ. 1) GO TO 1
      MODCLS=BITGET(ISUBLT(3,LC2),10,4)
    1 IF(IBLK .EQ. 0) GO TO 120
      IF(IDTBL(1,IBLK) .EQ. BLANK) GO TO 3
      LISTLC=BITGET(IDTBL(3,IBLK),32,9)
      KLAS=BITGET(ISUBLT(3,LISTLC),10,4)
      ISZ=BITGET(ISUBLT(3,LISTLC),32,15)
      IF(IDTBL(5,IBLK) .NE. ISZ) GO TO 70
      GO.TO 5
    3 IF(MODCLS .NE. 1 .AND. MODCLS .NE. 2) GO TO 5
      IBNKSZ=BITGET(ISUBLT(3,LC2),32,15)
      IF(IDTBL(5,IBLK) .NE. IBNKSZ)CALL ERROR(58,IBKCOM,BLANK,IDM3,IDM4)
    5 NBLOC=0
      ISUM=0
      NTP=0
      TP=0
      ICOMST=IDTBL(8,IBLK)
      LOC=ICOMST
   10 PREVTP=TP
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 15
      TP=1
      IFST=BITGET(IDTBL(1,LOC),8,8)
      IF(IFST .LE. 213 .AND. IFST .GE. 201) TP=4
      GO TO 18
   15 TP=BITGET(IDTBL(3,LOC),10,3)
   18 SZ=1
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      IF(NDIM .EQ. 0) GO TO 22
      DO 20 I=1,NDIM
   20 SZ=SZ*IDTBL(4+I,LOC)
   22 IF(TP .NE. 2 .AND. TP .NE. 3) GO TO 25
      IF(MOD(ISUM,2) .NE. 0) CALL ERROR(64,IDTBL(1,LOC),IDTBL(2,LOC),
     $ IDTBL(1,IBLK),IDTBL(2,IBLK))
      ISUM=ISUM+SZ
```

```
25 ISUM=ISUM+SZ
   IF(IBLKDT .EQ. 1 .AND. IORD(PREVTP+1) .GT. IORD(TP+1))
  $ CALL ERROR(65,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)
   IF(KLAS .EQ. 10 .OR. IDTBL(1,IBLK) .EQ. BLANK) GO TO 38
   IF(KLAS .EQ. 9) GO TO 35
   IF(KLAS .EQ. 7) GO TO 40
   ICTGR2=1
   IF(TP .EQ. PREVTP) GO TO 35
   IF(PREVTP .EQ. 0) GO TO 32
   DO 30 I=1,NTP
   IF(TP .EQ. ITPS(I)) GO TO 110
30 CONTINUE
32 NTP=NTP+1
   ITPS(NTP)=TP
35 IF(IBLKDT .EQ. 1) GO TO 38
   IF(BITGET(IDTBL(3,LOC),21,1) .EQ. 0)
  $ CALL ERROR(75,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
38 LOC=IDTBL(8,LOC)
   IF(LOC .NE. ICOMST) GO TO 10
   GO TO 65
40 IF(TP .EQ. PREVTP) GO TO 45
   NBLOC=NBLOC+1
   CMBLK(1,NBLOC)=TP
   CMBLK(2,NBLOC)=0
45 CMBLK(2,NBLOC)=CMBLK(2,NBLOC)+SZ
   IF(IDTBL(1,IBLK) .NE. SES(1)) GO TO 85
   IF(IDTBL(2,IBLK) .NE. SES(2)) GO TO 85
   NSES=NSES+1
   IF(NSES .GT. 13) GO TO 80
   IF(IDTBL(1,LOC) .EQ. SESCOM(1,NSES) .AND. IDTBL(2,LOC) .EQ.
  $ SESCOM(2,NSES)) GO TO 85
80 SESERR=1
85 LOC=IDTBL(8,LOC)
   IF(LOC .NE. ICOMST) GO TO 10
   IPTR=ISUBLT(4,LISTLC)
   NDPTR=IPTR+NBLOC-1
   KOUNT=0
   NGRP=NBLOC
   DO 50 I=IPTR,NDPTR
   KOUNT=KOUNT+1
   IF(KOUNT .GT. NBLOC) GO TO 65
   SZ=BITGET(INTFAC(I),17,17)
   TP=BITGET(INTFAC(I),20,3)
   IF(TP .NE. 0) GO TO 48
   CMBLK(2,KOUNT)=CMBLK(2,KOUNT)-SZ
   IF(CMBLK(2,KOUNT) .EQ. 0) GO TO 50
   IF(CMBLK(2,KOUNT) .LT. 0) GO TO 90
   KOUNT=KOUNT-1
   NGRP=NGRP+1
   GO TO 50
48 IF(CMBLK(1,KOUNT) .NE. TP .OR. CMBLK(2,KOUNT) .NE. SZ) GO TO 90
50 CONTINUE
   IF(NGRP .NE. BITGET(ISUBLT(3,LISTLC),6,6)) GO TO 90
65 IBLK=IDTBL(4,IBLK)
   GO TO 1
   70 CALL ERROR(58,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)
      GO TO 65
   90 CALL ERROR(57,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)
      GO TO 65
  110 CALL ERROR(63,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)
      GO TO 65
C**'CHECK THAT COMMON BLOCK "SESCOM" IS WELL DEFINED
  120 IF(ICTGR2 .EQ. 0 .AND. (MODCLS .EQ. 1 .OR. MODCLS .EQ. 2))
     $ CALL ERROR(73,IDM1,IDM2,IDM3,IDM4)
      IF(NSES .EQ. 0) GO TO 130
      IF(SESERR .EQ. 1 .OR. NSES .LT. 13)
     $ CALL ERROR(48,IDM1,IDM2,IDM3,IDM4)
      RETURN
  130 CALL ERROR(66,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE COMEXT
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER BITGET
      ICOMLC=0
      ICOMNM=IDTBL(9,LOC)
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER BITGET
      ICOMLC=0
      ICOMNM=IDTBL(9,LOC)
      ICOMST=IDTBL(8,ICOMNM)
      ICOMND=IDTBL(9,ICOMNM)
      ICOMSZ=IDTBL(5,ICOMNM)
      NXTLOC=ICOMND
      DO 20 I=1,ICOMSZ
      NXTLOC=IDTBL(8,NXTLOC)
      MUL=1
      ISZ=1
      ITP=BITGET(IDTBL(3,NXTLOC),10,3)
      IF(ITP .EQ. 2 .OR. ITP .EQ. 3) MUL=2
      IF(LOC .EQ. NXTLOC) GO TO 25
      IF(BITGET(IDTBL(3,NXTLOC),1,1) .EQ. 1) GO TO 10
      GO TO 20
 10   NDIM=BITGET(IDTBL(3,NXTLOC),7,6)
      DO 15 J=1,NDIM
 15   ISZ=ISZ*IDTBL(J+4,NXTLOC)
 20   ICOMLC=ICOMLC+ISZ*MUL
 25   ILFT=ICOMLC
      IRHT=ICOMSZ-ICOMLC
      NXTLOC=LOC
      IOFFST=IDTBL(11,NXTLOC)
 30   NXTLOC=IDTBL(10,NXTLOC)
      IF(NXTLOC .EQ. LOC) RETURN
      IOFF2=IDTBL(11,NXTLOC)
      IF((IOFFST-IOFF2) .GT. ILFT) GO TO 50
      ITP=BITGET(IDTBL(3,NXTLOC),10,3)
      ISZ=1
      IF(ITP .NE. 2 .AND. ITP .NE. 3) GO TO 32
      IF(MOD((ILFT-IOFFST+IOFF2),2) .NE. 0) CALL ERROR(64,IDTBL(1,NXTLOC
     $),IDTBL(2,NXTLOC),IDTBL(1,ICOMNM),IDTBL(2,ICOMNM))
      ISZ=2
 32   IF(BITGET(IDTBL(3,NXTLOC),1,1) .NE. 1) GO TO 40
      NDIM=BITGET(IDTBL(3,NXTLOC),7,6)
      DO 35 I=1,NDIM
 35   ISZ=ISZ*IDTBL(4+I,NXTLOC)
 40   IF((ISZ-(IOFFST-IOFF2)) .GT. IRHT) GO TO 50
      GO TO 30
 50   CALL ERROR(47,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE COMSCH
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      J=INITID(3)
      IF(J .EQ. 0) GO TO 15
      DO 10 I=1,NID
      IF(ICOMP(NXTID,IDTBL,J,11) .EQ. 0) GO TO 5
      ISRCH(3)=1
      LOC=J
      RETURN
    5 J=IDTBL(4,J)
      IF(J .EQ. 0) GO TO 15
   10 CONTINUE
   15 ISRCH(3) =0
      RETURN
      END
      SUBROUTINE CTGOTO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(4)
      INTEGER STATRA,A,BLANK,RPAR,COMMA
      INTEGER BITPUT,BITGET
      DATA BLANK/1H /,COMMA/1H,/,LPAR/1H(/,RPAR/1H)/
      DATA IALPH/1HG,1HO,1HT,1HO/
      DO 5 I=1,4
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30
    5 CONTINUE
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30
      NBLOCK=NBLOCK+1
      JBLOCK=NBLOCK
      NBRNCH=0
   10 CALL GNLE
      IF(JTYP .NE. 5) GO TO 30
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      IF(NBRNCH .EQ. 0) GO TO 15
      DO 12 I=1,NBRNCH
      IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 17
   12 CONTINUE
   15 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=LOC
      NBRNCH=NBRNCH+1
   17 IF(NEXT(JPTR) .EQ. COMMA) GO TO 10
      IF(A(JPTR-1) .NE. RPAR) GO TO 30
      IF(NEXT(JPTR) .NE. COMMA) GO TO 30
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 30
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 20
      IDTYP=1
      CALL STORE
      LOC=NID
   20 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(NEXT(JPTR) .NE. BLANK) GO TO 30
      IBLOCK(JBLOCK)=2000+LOC
      NB=1
      RETURN
   30 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE DATA
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH(4)
      INTEGER A,RPAR,COMMA,SLASH,BLANK,ASTRIK,PLUS
      INTEGER BITPUT,BITGET
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,SLASH/1H//,BLANK/1H /,
     * ASTRIK/1H*/,PLUS/1H+/,MINUS/1H-/
      DATA IALPH/1HD,1HA,1HT,1HA/
      DO 5 I=1,4
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60
    5 CONTINUE
    6 LST1S7=0
      LST2SZ=0
    8 ISZ=1
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 60
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 9
      IDTYP=1
      CALL STORE
      LOC=NID
    9 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)
     $ CALL ERROR(30,NXTID(1),NXTID(2),IDM3,IDM4)
      CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 0) GOTO 10
      ICOMLC=IDTBL(9,LOC)
      IF(IBLKDT .EQ. 0 .OR. IDTBL(1,ICOMLC) .EQ. BLANK)
     $ CALL ERROR(28,NXTID(1),NXTID(2),IDM3,IDM4)
   10 IF(NEXT(JPTR) .NE. LPAR) GO TO 25
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 90
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      I=0
   15 I=I+1
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 60
      IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(N2 .GT. IDTBL(4+I,LOC)) CALL ERROR(18,IDM1,IDM2,IDM3,IDM4)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 15
      IF(I .NE. NDIM) GO TO 80
      IF(A(JPTR-1) .NE. RPAR) GO TO 80
      GO TO 35
   25 JPTR=JPTR-1
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 35
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      DO 30 I=1,NDIM
   30 ISZ=ISZ*IDTBL(4+I,LOC)
      IF(BITGET(IDTBL(3,LOC),14,1) .EQ. 1)
     $ CALL ERROR(29,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
   35 LST1SZ=LST1SZ+ISZ
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,14)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 8
      IF(A(JPTR-1) .NE. SLASH) GO TO 60
```

```
40 NRPEAT=1
   CALL GNLE
   IF(JTYP .EQ. 3) GO TO 47
   IF(JTYP .NE. 5) GO  TO 45
   IF(NEXT(JPTR) .NE. ASTRIK) GO TO 50
   NRPEAT=N2
   CALL GNLE
45 IF(A(JPTR-1) .NE. PLUS .AND. A(JPTR-1) .NE. MINUS) GO TO 47
   CALL GNLE
47 KK=NEXT(JPTR)
50 IF(JTYP .GE. 3.AND. JTYP .LE. 6) GO TO 55
   IF(JTYP .EQ. 7 .AND. LOGID .EQ. 10) GO TO 55
   IF(JTYP .EQ. 7 .AND. LOGID .EQ. 11) GO TO 55
   GO TO 70
55 LST2SZ=LST2SZ+NRPEAT
   IF(A(JPTR-1) .EQ. COMMA) GO TO 40
   IF(A(JPTR-1) .NE. SLASH) GO TO 60
   IF(LST1SZ .NE. LST2SZ) CALL ERROR(31,IDM1,IDM2,IDM3,IDM4)
   IF(NEXT(JPTR) .EQ. COMMA) GO TO 6
   IF(A(JPTR-1) .NE. BLANK) GO TO 60
   RETURN
60 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
   RETURN
70 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)
   RETURN
80 CALL ERROR(19,IDM1,IDM2,IDM3,IDM4)
   RETURN
90 CALL ERROR(13,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
   RETURN
   END
```

```
      SUBROUTINE DESCRP
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP
      DIMENSION FORMT(7)
      INTEGER A,FORMT,DECPT,PEE,EX
      DATA FORMT/1HF,1HE,1HG,1HD,1HI,1HL,1HA/
      DATA DECPT/1H./,PEE/1HP/,EX/1HX/,MINUS/1H-/
      ISCLFC=0
      INT=0
      IMINUS=0
      IDES=1
      JPTR=IDESST
      IF(NEXT(JPTR) .NE. MINUS) GO TO 5
      IMINUS=1
      GO TO 6
    5 JPTR=IDESST
    6 CONTINUE
      CALL GNLE
      IF(JTYP .EQ. 3) GO TO 80
      IF(JTYP .EQ. 5) GO TO 10
      IF(JTYP .NE. 2) GO TO 15
      IF(NXTID(1) .EQ. PEE .AND. ISCLFC .EQ. 0) GO TO 20
      IF(INT .EQ. 1 .AND. N2 .LT. 1) GO TO 15
      IF(ISCLFC .EQ. 0 .AND. IMINUS .EQ. 1) GO TO 15
      GO TO 25
   10 INT=1
      GO TO 6
   15 IDES=0
      RETURN
   20 IF(INT .EQ. 0) GO TO 15
      ISCLFC=1
      INT=0
      GO TO 6
   25 DO 30 I=1,7
      IF(NXTID(1) .EQ. FORMT(I)) GO TO 45
   30 CONTINUE
      IF(NXTID(1) .NE. EX .OR. ISCLFC .EQ. 1 .OR. INT .EQ. 0) GO TO 15
      GO TO 80
   45 CALL GNLE
      IF(JTYP .NE. 5) GO TO 15
      NWIDTH=N2
      IF(I .LE. 4) GO TO 60
      IF(ISCLFC .EQ. 1 .OR. NWIDTH .LT. 1) GO TO 15
      IF(I .EQ. 7 .AND. NWIDTH .GT. 4) GO TO 15
      IDESND=JPTR-1
      RETURN
   60 IF(NEXT(JPTR) .NE. DECPT) GO TO 15
      IF(NWIDTH .LT. 2) GO TO 15
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 15
      NDCPLS=N2
      IDESND=JPTR-1
      IF(I .EQ. 1) GO TO 65
      IF(NWIDTH .LT. (NDCPLS+6)) GO TO 15
      RETURN
   65 IF(NWIDTH .LT. NDCPLS) GO TO 15
      RETURN
   80 IDESND=JPTR-1
      RETURN
      END
      SUBROUTINE DIMEN
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH(9),IDIM(3)
      INTEGER A,D,RPAR,COMMA
      INTEGER BITPUT,BITGET,COMLOC
      DATA IALPH/1HD,1HI,1HM,1HE,1HN,1HS,1HI,1HO,1HN/
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/
      DO 10 I=1,9
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 110
   10 CONTINUE
   12 CALL GNLE
      IF(JTYP .NE. 2) GO TO 110
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 5
      IDTYP=1
      CALL STORE
      LOC=NID
    5 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)
     $ CALL ERROR(11,NXTID(1),NXTID(2),IDM3,IDM4)
      CALL IMPTYP
      IDTBL(3,LOC)=HITPUT(IDTBL(3,LOC),1,1)
      IE=LOC
      IF(NEXT(JPTR) .NE. LPAR) GO TO 110
      INCR=1
      I=0
   15 I=I+1
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 13
      IDIM(I)=N2
      IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      INCR=INCR*N2
      GO TO 14
   13 IF(JTYP .NE. 2) GO TO 110
      IDTYP=1
      CALL SEARCH
      IF(ISRCH(2) .NE. 0) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GOTO 25
      IDTYP=1
      CALL STORE
      LOC=NID
   25 IF(BITGET(IDTHL(3,LOC),12,1) .NE. 1)
     $ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 120
      IDTBL(3,LOC)=HITPUT(IDTBL(3,LOC),1,13)
      CALL IMPTYP
      IF(BITGET(IDTHL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)
      IDIM(I)=2**17+LOC
   14 IF(NEXT(JPTR) .EQ. COMMA) GO TO 15
      IF(A(JPTR-1) .NE. RPAR) GO TO 110
      LOC=IE
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)
      IF(I .GT. 3) GO TO 110
      DO 30 J=1,I
   30 IDTBL(4+J,LOC)=IDIM(J)
      IF(BITGET(IDTBL(3,LOC),16,1) .NE. 1) GO TO 50
      COMLOC=IDTBL(9,LOC)
      IT=1
      ITP=BITGET(IDTBL(3,LOC),10,3)
      IF(ITP .EQ. 2 .OR. ITP .EQ. 3) IT=2
      IDTBL(5,COMLOC)=IDTBL(5,COMLOC)+IT*(INCR-1)
   50 CONTINUE
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 12
      IF(JPTR .GT. N) RETURN
  110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
  120 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
      END
      SUBROUTINE DO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION PARAM(3)
      INTEGER A,BLANK,COMMA,EQUALS,DEE,OH,PARAM,STATRA
      INTEGER BITPUT,BITGET
      DATA BLANK/1H /,COMMA/1H,/,EQUALS/1H=/,DEE/1HD/,OH/1HO/
      IF(NEXT(JPTR) .NE. DEE) GO TO 50
      IF(NEXT(JPTR) .NE. OH) GO TO 50
      CALL GNLE
      IF(JTYP .NE. 5) GOTO 50
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,15)
      IF(IOVFLW .EQ. 1) GO TO 2
      NSTACK=NSTACK+1
      IF(NSTACK .GT. 50) GO TO 1
      ISTACK(1,NSTACK)=LOC
      ISTACK(2,NSTACK)=0
      ISTACK(3,NSTACK)=ILOOP
      ILOOP=NSTACK
      GO TO 2
    1 IOVFLW=1
      WRITE(6,60)
   60 FORMAT(///5X,50H DO STACK OVERFLOW - DO LOOP PROCESSING TERMINATED
     *)
    2 CALL GNLE
      IF(JTYP .NE. 2) GO TO 50
      CALL SEARCH
      IF(ISPCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 5
      IDTYP=1
      CALL STORE
      LOC=NID
    5 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(40,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(NEXT(JPTR) .NE. EQUALS) GO TO 50
      IF(IOVFLW .EQ. 1) GO TO 8
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=3000+LOC
      ISTACK(4,NSTACK)=LOC
    8 PARAM(3)=1
      DO 30 I=1,3
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 10
      PARAM(I)=N2
      IF(N2 .LE. 0) CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)
      GO TO 20
```

```
10 IF(JTYP .NE. 2) GO TO 50
   CALL SEARCH
   IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
   IF(ISRCH(1) .EQ. 1) GO TO 15
   IDTYP=1
   CALL STORE
   LOC=NID
15 CALL IMPTYP
   IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
  $ CALL ERROR(40,NXTID(1),NXTID(2),IDM3,IDM4)
   IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
  $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
   NBLOCK=NBLOCK+1
   IBLOCK(NBLOCK)=7000+LOC
   IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ILOOP,32)
   PARAM(I)=0
20 IF(I .EQ. 3) GO TO 30
   IF(I .EQ. 1) GO TO 25
   IF(NEXT(JPTR) .EQ. BLANK) GO TO 35
   JPTR=JPTR-1
25 IF(NEXT(JPTR) .NE. COMMA) GO TO 50
30 CONTINUE
   IF(NEXT(JPTR) .NE. BLANK) GO TO 50
35 IF(PARAM(1) .EQ. 0 .OR. PARAM(2) .EQ. 0) GO TO 40
   IF(PARAM(2) .LT. PARAM(1)) CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)
   IF(PARAM(3) .EQ. 0) GO TO 40
   IF((PARAM(2)+PARAM(3)-1) .GT. 2**17-1)
  $ CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)
40 NBLOCK=NBLOCK+1
   IBLOCK(NBLOCK)=998
   NBRNCH=1
   NB=1
   RETURN
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
   RETURN
   END
```

```
      SUBROUTINE EQUIV
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH(11),IDIM(3)
      INTEGER BITPUT,BITGET
      INTEGER A,RPAR,COMMA,BOFFST,BLANK
      DATA IALPH/1HE,1HQ,1HU,1HI,1HV,1HA,1HL,1HE,1HN,1HC,1HE/
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /
      DO 5 I=1,11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 130
    5 CONTINUE
    8 IF(NEXT(JPTR) .NE. LPAR) GO TO 130
      LSTLOC=0
      BOFFST=0
      J=0
 .120 J=J+1
      CALL GNLE
      ILOC=1
      IF(JTYP .NE. 2) GO TO 130
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 9
      IDTYP=1
      CALL STORE
      LOC=NID
    9 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)
     $ CALL ERROR(20,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) GO TO 150
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      DO 10 I=1,NDIM
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 130
      IDIM(I)=N2
      IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(N2 .GT. IDTBL(4+I,LOC)) CALL ERROR(18,IDM1,IDM2,IDM3,IDM4)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 10
      IF(A(JPTR-1) .NE. RPAR) GO TO 130
      GO TO 15
   10 CONTINUE
      GO TO 140
   15 NDIM=I
      ILOC=IDIM(1)
      IF(NDIM .EQ. 1) GO TO 25
      ILOC=ILOC+(IDIM(2)-1)*IDTBL(5,LOC)
      IF(NDIM .EQ. 2) GO TO 25
      ILOC=ILOC+(IDIM(3)-1)*IDTBL(5,LOC)*IDTBL(6,LOC)
   25 IT=BITGET(IDTBL(3,LOC),10,3)
      IF(IT .EQ. 2 .OR. IT .EQ. 3) ILOC=2*ILOC
      IOFFST=1-ILOC-BOFFST
      GO TO 45
   30 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
```

```
      IOFFST=BOFFST
      JPTR=JPTR-1
   45 IF(BITGET(IDTBL(3,LOC),17,1) .EQ. 1) GO TO 57
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,17)
      IF(LSTLOC .EQ. 0) GO TO 50
      IDTBL(10,LSTLOC)=LOC
      GO TO 55
   50 IFSTLC=LOC
   55 IDTBL(11,LOC)=IOFFST
      LSTLOC=LOC
      IF(J .NE. 1) GO TO 100
      GO TO 98
   57 LOC3=LOC
   58 LOC3=IDTBL(10,LOC3)
      IF(LOC3 .EQ. 0) GO TO 59
      IF(LOC3 .EQ. LOC) GO TO 60
      GO TO 58
   59 JLOC=ILOC+IDTBL(11,LOC)
      IF(JLOC .NE. IDIS) CALL ERROR(20,IDTBL(1,LOC),IDTBL(2,LOC),
     $ IDM3,IDM4)
      GO TO 100
   60 IF(LSTLOC .NE. 0) GO TO 63
      IFSTLC=LOC
      GO TO 65
   63 IDTBL(10,LSTLOC)=LOC
   65 LOC2=LOC
   70 NXTLOC=IDTBL(10,LOC2)
      IF(NXTLOC .EQ. LOC) GO TO 75
      LOC2=NXTLOC
      GO TO 70
   75 IDTBL(10,LOC2)=0
      LSTLOC=LOC2
      IOFFDF=IOFFST-IDTBL(11,LOC)
      IF(IOFFDF) 80,98,90
   80 LOC2=LOC
   85 IDTBL(11,LOC2)=IDTBL(11,LOC2)+IOFFDF
      LOC2=IDTBL(10,LOC2)
      IF(LOC2 .EQ. 0) GO TO 98
      GO TO 85
   90 LOC2=IFSTLC
   95 IF(LOC2 .EQ. LOC) GO TO 97
      IDTBL(11,LOC2)=IDTBL(11,LOC2)-IOFFDF
      LOC2=IDTBL(10,LOC2)
      GO TO 95
   97 BOFFST=BOFFST-IOFFDF
   98 IDIS=ILOC+IDTBL(11,LOC)
  100 IF(NEXT(JPTR) .EQ. COMMA) GO TO 120
      IF(A(JPTR-1) .NE. RPAR) GO TO 130
      IF(J .EQ. 1) GO TO 130
      IDTBL(10,LSTLOC)=IFSTLC
      JK=0
      LOC3=IFSTLC
      I=0
  110 I=I+1
      LOC3=IDTBL(10,LOC3)
      IF(BITGET(IDTBL(3,LOC3),16,1) .EQ. 0) GO TO 105
      JK=JK+1
      IF(JK .GT. 1) CALL ERROR(21,IDTBL(1,LOC3),IDTBL(2,LOC3),
     $ IDTBL(1,LOC1),IDTBL(2,LOC1))
      LOC1=LOC3
      LOC=LOC3
      CALL COMEXT
  105 IF(LOC3 .NE. IFSTLC) GO TO 110
      IF(NEXT(JPTR) .EQ. BLANK) RETURN
      IF(A(JPTR-1) .EQ. COMMA) GO TO 8
  130 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
  140 CALL ERROR(19,IDM1,IDM2,IDM3,IDM4)
      RETURN
  150 CALL ERROR(13,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
      END
      SUBROUTINE ERROR(IERROR,IDM1,IDM2,IDM3,IDM4)
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FLOW/IFL,IRP
      WRITE(6,1)
    1 FORMAT(1X,100H****************************************************
     1************************************************)
      GO TO (5,15,25,35,45,55,65,75,85,95,105,115,125,135,145,155,165,
     A175,185,195,205,215,225,235,245,255,265,275,285,295,305,315,325,
     * 335,345,355,365,375,385,395,405,415,425,435,445,455,465,475,485,
     * 495,505,515,525,535,545,555,565,575,585,595,605,615,625,635,645,
     $ 655,665,675,685,695,705,715,725,735,745,755,765,775,785,795,805,
     $ 815,825,835,845,855,865,875,885,895,905,915,925,935,945),IERROR
    5 WRITE(6, 10)
   10 FORMAT(6X,26H THIS STATEMENT IS ILLEGAL)
      GO TO 1000
   15 WRITE(6, 20)
   20 FORMAT(6X,31H THIS STATEMENT IS OUT OF ORDER)
      GO TO 1000
   25 WRITE(6, 30)
   30 FORMAT(6X,39H VALUE OF INTEGER CONSTANT IS TOO LARGE)
      GO TO 1000
   35 WRITE(6, 40)
   40 FORMAT(6X,28H TOO MANY CONTINUATION CARDS)
      GO TO 1000
   45 WRITE(6, 50)
   50 FORMAT(6X,30H HOLLERITH STRING IS TOO LARGE)
      GO TO 1000
   55 WRITE(6, 60)
   60 FORMAT(6X,26H VARIABLE NAME IS TOO LONG)
      GO TO 1000
   65 WRITE(6, 70)
   70 FORMAT(6X,31H SYNTAX ERROR IN THIS STATEMENT)
      IF(ITYP .LE. 18 .AND. IFL .GT. 0) IFL=-1
      GO TO 1000
   75 WRITE(6, 80)
   80 FORMAT(6X,46H ARRAY DIMENSION IS OUTSIDE OF ALLOWABLE RANGE)
      GO TO 1000
   85 WRITE(6, 90)
   90 FORMAT(6X,45H ILLEGAL VARIABLE DIMENSION IN THIS STATEMENT)
      GO TO 1000
   95 WRITE(6,100) IDM1,IDM2
  100 FORMAT(6X,33H THE FUNCTION OR SUBROUTINE NAME ,A4,A2,18H IS USED I
     $LLEGALLY)
      GO TO 1000
  105 WRITE(6,110) IDM1,IDM2
  110 FORMAT(6X,14H THE VARIABLE ,A4,A2,32H HAS BEEN PREVIOUSLY DIMENSIO
     $NED)
      GO TO 1000
  115 WRITE(6,120) IDM1,IDM2
  120 FORMAT(6X,14H THE VARIABLE ,A4,A2,26H HAS BEEN PREVIOUSLY TYPED)
      GO TO 1000
  125 WRITE(6,130) IDM1,IDM2
  130 FORMAT(6X,14H THE VARIABLE ,A4,A2,38H IS ILLEGALLY FOLLOWED BY A L
     $EFT PAREN)
      GO TO 1000
  135 WRITE(6,140) IDM1,IDM2
  140 FORMAT(6X,26H THE DIMENSIONED VARIABLE ,A4,A2,18H IS USED ILLEGALL
     $Y)
      GO TO 1000
  145 WRITE(6,150) IDM1
  150 FORMAT(6X,18H STATEMENT NUMBER ,I5,15H IS NOT DEFINED)
      IF(IFL .GT. 0) IFL=-1
      GO TO 1000
  155 WRITE(6,160) IDM1
  160 FORMAT(6X,18H STATEMENT NUMBER ,I5,18H IS NOT REFERENCED)
      GO TO 1000
  165 WRITE(6,170) IDM1,IDM2
  170 FORMAT(6X,18H ILLEGAL VARIABLE ,A4,A2,10H IN COMMON)
      GO TO 1000
  175 WRITE(6,180)
  180 FORMAT(6X,43H VALUE OF ARRAY SUBSCRIPT EXCEEDS DIMENSION)
      GO TO 1000
  185 WRITE(6,190)
  190 FORMAT(6X,25H ERROR IN ARRAY SUBSCRIPT)
      GO TO 1000
  195 WRITE(6,200) IDM1,IDM2
  200 FORMAT(6X,18H ILLEGAL VARIABLE ,A4,A2,16H IS EQUIVALENCED)
      GO TO 1000
  205 WRITE(6,210) IDM1,IDM2,IDM3,IDM4
  210 FORMAT(6X,22H THE COMMON VARIABLES ,A4,A2,5H AND ,A4,A2,17H ARE EQ
     $UIVALENCED)
      GO TO 1000
  215 WRITE(6,220)
  220 FORMAT(6X,19H ILLEGAL I/O DEVICE)
      GO TO 1000
  225 WRITE(6,230)
  230 FORMAT(6X,37H ILLEGAL CHARACTER IN THIS EXPRESSION)
      GO TO 1000
  235 WRITE(6,240) IDM1,IDM2
  240 FORMAT(6X,25H ILLEGAL SUBROUTINE NAME ,A4,A2)
      GO TO 1000
  245 WRITE(6,250)
  250 FORMAT(6X,50H SUBROUTINE TABLE OVERFLOW - PROCESSING TERMINATED)
      GO TO 1000
  255 WRITE(6,260) IDM1,IDM2
  260 FORMAT(6X,77H INCORRECT NUMBER OF ARGUMENTS IN CALLING SEQUENCE OF
     $ FUNCTION OR SUBROUTINE ,A4,A2)
      GO TO 1000
  265 WRITE(6,270)
  270 FORMAT(6X,19H ILLEGAL ASSIGNMENT)
      GO TO 1000
  275 WRITE(6,280) IDM1,IDM2
  280 FORMAT(6X,14H THE VARIABLE ,A4,A2,42H APPEARS IN A DATA STATEMENT
     $AND IN COMMON)
      GO TO 1000
  285 WRITE(6,290) IDM1,IDM2
  290 FORMAT(6X,14H THE VARIABLE ,A4,A2,44H HAS PREVIOUSLY APPEARED IN A
     $ DATA STATEMENT)
```

```
      GO TO 1000
  295 WRITE(6,300) IDM1,IDM2
  300 FORMAT(6X,22H THE FORMAL PARAMETER ,A4,A2,31H APPEARS IN THIS DATA
     $ STATEMENT)
      GO TO 1000
  305 WRITE(6,310)
  310 FORMAT(6X,24H LIST SIZES DO NOT MATCH)
      GO TO 1000
  315 WRITE(6,320)
  320 FORMAT(6X,24H ILLEGAL STATEMENT LABEL)
      IF(IFL .GT. 0) IFL=-1
      GO TO 1000
  325 WRITE(6,330)
  330 FORMAT(6X,26H DUPLICATE STATEMENT LABEL)
      IF(IFL .GT. 0) IFL=-1
      GO TO 1000
  335 WRITE(6,340)
  340 FORMAT(6X,34H THIS STATEMENT CAN NOT BE REACHED)
      GO TO 1000
  345 WRITE(6,350)
  350 FORMAT(6X,31H DO LOOPS ARE IMPROPERLY NESTED)
      GO TO 1000
  355 WRITE(6,360)
  360 FORMAT(6X,32H FORMAT STATEMENT IS NOT LABELED)
      GO TO 1000
  365 WRITE(6,370)
  370 FORMAT(6X,20H ILLEGAL DO TERMINAL)
      GO TO 1000
  375 WRITE(6,380)
  380 FORMAT(6X,37H LAST EXECUTABLE STATEMENT IS ILLEGAL)
      IF(IFL .GT. 0) IFL=-1
      GO TO 1000
  385 WRITE(6,390) IDM1,IDM2
  390 FORMAT(6X,24H THE VARIABLE REFERENCE ,A4,A2,18H IS NOT AN INTEGER)
      GO TO 1000
  395 WRITE(6,400) IDM1,IDM2
  400 FORMAT(6X,27H THE DO PARAMETER OR INDEX ,A4,A2,18H IS NOT AN INTEG
     $ER)
      GO TO 1000
  405 WRITE(6,410)
  410 FORMAT(6X,52H VALUE OF DO PARAMETER IS OUTSIDE OF ALLOWABLE RANGE)
      GO TO 1000
  415 WRITE(6,420)
  420 FORMAT(6X,32H COMPLEX EXPRESSIONS ARE ILLEGAL)
      GO TO 1000
  425 WRITE(6,430)
  430 FORMAT(6X,24H ILLEGAL VARIABLE FORMAT)
      GO TO 1000
  435 WRITE(6,440)
  440 FORMAT(6X,39H THIS STATEMENT SHOULD HAVE AN I/O LIST)
      GO TO 1000
  445 WRITE(6,450)
  450 FORMAT(6X,50H STATEMENT FOLLOWING LOGICAL EXPRESSION IS ILLEGAL)
      GO TO 1000
  455 WRITE(6,460)
```

```
460   FORMAT(6X,44H REAL NUMBER LIES OUTSIDE OF ALLOWABLE RANGE)
      GO TO 1000
465   WRITE(6,470)
470   FORMAT(6X,42H THIS EQUIVALENCE STATEMENT EXTENDS COMMON)
      GO TO 1000
475   WRITE(6,480)
480   FORMAT(6X,40H ILLEGAL VARIABLE IN COMMON BLOCK SESCOM)
      GO TO 1000
485   WRITE(6,490) IDM1,IDM2
490   FORMAT(6X,12H SUBPROGRAM ,A4,A2,19H HAS INCORRECT TYPE)
      GO TO 1000
495   WRITE(6,500) IDM1
500   FORMAT(6X,23H WARNING - ARGUMENT NO.,I3,34H MAY HAVE INCORRECT DIM
     *ENSIONALITY)
      GO TO 1000
505   WRITE(6,510) IDM1
510   FORMAT(6X,13H ARGUMENT NO.,I3,19H HAS INCORRECT TYPE)
      GO TO 1000
515   WRITE(6,520)
520   FORMAT(6X,49H WARNING - THIS MODULE IS NOT IN THE SESCOMP LIST)
      GO TO 1000
525   WRITE(6,530) IDM1,IDM2
530   FORMAT(6X,14H THE VARIABLE ,A4,A2,29H PREVIOUSLY APPEARS IN COMMON
     $)
      GO TO 1000
535   WRITE(6,540) IDM1
540   FORMAT(6X,13H ARGUMENT NO.,I3,11H IS INVALID)
      GO TO 1000
545   WRITE(6,550) IDM1
550   FORMAT(6X,13H ARGUMENT NO.,I3,29H IS DESIGNATED LOGICAL OUTPUT)
      GO TO 1000
555   WRITE(6,560) IDM1,IDM2
560   FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A4,A2)
      GO TO 1000
565   WRITE(6,570) IDM1,IDM2
570   FORMAT(6X,41H WARNING - VARIABLE TYPE IN COMMON BLOCK ,A4,A2,41H D
     $OES NOT AGREE WITH INTERFACE DEFINITION)
      GO TO 1000
575   WRITE(6,580) IDM1,IDM2
580   FORMAT(6X,14H COMMON BLOCK ,A4,A2,19H HAS INCORRECT SIZE)
      GO TO 1000
585   WRITE(6,590)
590   FORMAT(6X,58H EXTERNAL REFERENCE TABLE OVERFLOW - PROCESSING TERMI
     $NATED)
      GO TO 1000
595   WRITE(6,600)
600   FORMAT(6X,52H COMMON BLOCK TABLE OVERFLOW - PROCESSING TERMINATED)
      GO TO 1000
605   WRITE(6,610) IDM1,IDM2
610   FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A4,A2)
      GO TO 1000
615   WRITE(6,620) IDM1,IDM2
620   FORMAT(6X,14H COMMON BLOCK ,A4,A2,27H IS NOT IN THE SESCOMP LIST)
      GO TO 1000
625   WRITE(6,630) IDM1,IDM2
```

```
630 FORMAT(6X,25H CATEGORY 2 COMMON BLOCK ,A4,A2,23H IS NOT GROUPED BY
   $ TYPE)
    GO TO 1000
635 WRITE(6,640) IDM1,IDM2,IDM3,IDM4
640 FORMAT(6X,38H DOUBLE PRECISION OR COMPLEX VARIABLE ,A4,A2,56H DOES
   $ NOT BEGIN ON AN EVEN LOCATION WITHIN COMMON BLOCK ,A4,A2)
    GO TO 1000
645 WRITE(6,650) IDM1,IDM2
650 FORMAT(6X,26H VARIABLE IN COMMON BLOCK ,A4,A2,16H IS OUT OF ORDER)
    GO TO 1000
655 WRITE(6,660)
660 FORMAT(6X,56H THE COMMON BLOCK SESCOM DOES NOT APPEAR IN THIS PROG
   $RAM)
    GO TO 1000
665 WRITE(6,670) IDM1,IDM2
670 FORMAT(6X,14H THE DO INDEX ,A4,A2,13H IS REDEFINED)
    RETURN
675 WRITE(6,680) IDM1,IDM2
680 FORMAT(6X,24H THE VARIABLE DIMENSION ,A4,A2,13H IS REDEFINED)
    RETURN
685 WRITE(6,690) IDM1,IDM2
690 FORMAT(6X,23H THE ASSIGNED VARIABLE ,A4,A2,24H IS ILLEGALLY REFERE
   $NCED)
    RETURN
695 WRITE(6,700) IDM1,IDM2
700 FORMAT(6X,14H THE VARIABLE ,A4,A2,30H IS REFERENCED BUT NOT DEFINE
   $D)
    RETURN
705 WRITE(6,710) IDM1,IDM2
710 FORMAT(6X,14H THE VARIABLE ,A4,A2,45H IS REFERENCED ILLEGALLY BY A
   $N ASSIGNED GO TO)
    RETURN
715 WRITE(6,720) IDM1,IDM2
720 FORMAT(6X,18H THE DO PARAMETER ,A4,A2,13H IS REDEFINED)
    RETURN
725 WRITE(6,730)
730 FORMAT(6X,49H THIS MODULE CONTAINS NO CATEGORY 2 COMMON BLOCKS)
    GO TO 1000
735 WRITE(6,740) IDM1,IDM2
740 FORMAT(6X,24H THE ANSI FUNCTION NAME ,A4,A2,27H IS MISUSED IN THIS
   $ PROGRAM)
    GO TO 1000
745 WRITE(6,750) IDM1,IDM2
750 FORMAT(6X,14H THE VARIABLE ,A4,A2,60H APPEARS IN A CATEGORY 2 OR 3
   $ COMMON BLOCK BUT IS NEVER USED)
    GO TO 1000
755 WRITE(6,760)
760 FORMAT(6X,75H ARRAY SUBSCRIPT OR IMPLIED DO PARAMETER MAY LIE OUTS
   $IDE OF ALLOWABLE RANGE)
    GO TO 1000
765 WRITE(6,770) IDM1,IDM2,IDM3,IDM4
770 FORMAT(6X,22H MIXED MODE COMBINING ,A4,A2,6H WITH ,A4,A2)
    GO TO 1000
775 WRITE(6,780) IDM1
780 FORMAT(6X,33H INCORRECT EXPONENT AT CHAR. NO. ,I3)
```

```
      GO TO 1000
  785 WRITE(6,790) IDM1
  790 FORMAT(6X,47H VAR-CONST CONFUSION IN SUBSCRIPT AT CHAR. NO. ,I3)
      GO TO 1000
  795 WRITE(6,800) IDM1,IDM2,IDM3
  800 FORMAT(6X,40H SUBSCRIPT CONSTANT OR VARIABLE OF TYPE ,A4,A2,14H AT
     $ CHAR. NO. ,I3)
      GO TO 1000
  805 WRITE(6,810) IDM1
  810 FORMAT(6X,52H TOO MANY SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,
     $I3)
      GO TO 1000
  815 WRITE(6,820) IDM1
  820 FORMAT(6X,51H TOO FEW SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,
     $I3)
      GO TO 1000
  825 WRITE(6,830) IDM1
  830 FORMAT(6X,52H ILLEGAL TYPE IN RELATIONAL EXPRESSION AT CHAR. NO. ,
     $I3)
      GO TO 1000
  835 WRITE(6,840) IDM1
  840 FORMAT(6X,40H TOO MANY ARGUMENTS IN CALLING SEQUENCE ,I3)
      GO TO 1000
  845 WRITE(6,850)
  850 FORMAT(6X,41H TOO MANY FUNCTION REFS IN THIS STATEMENT)
      GO TO 1000
  855 WRITE(6,860) IDM1,IDM2
  860 FORMAT(6X,26H INVALID FORMAL PARAMETER ,A4,A2)
      GO TO 1000
  865 WRITE(6,870) IDM1,IDM2
  870 FORMAT(6X,19H THE FUNCTION NAME ,A4,A2,33H MAY HAVE BEEN PREVIOUSL
     $Y MISUSED)
      GO TO 1000
  875 WRITE(6,880) IDM1
  880 FORMAT(6X,39H ILLEGAL FIELD DESCRIPTOR AT CHAR. NO. ,I4)
      GO TO 1000
  885 WRITE(6,890)
  890 FORMAT(6X,38H TOO MANY FUNCTION DEFINING STATEMENTS)
      GO TO 1000
  895 WRITE(6,900)
  900 FORMAT(6X,71H TOO MANY EXTERNAL REFERENCES IN THIS STATEMENT - PRO
     $CESSING TERMINATED)
      GO TO 1000
  905 WRITE(6,910)
  910 FORMAT(6X,46H STATEMENT IS TOO LONG - PROCESSING TERMINATED)
      GO TO 1000
  915 WRITE(6,920)
  920 FORMAT(6X,46H SESCOMP LIST OVERFLOW - PROCESSING TERMINATED)
      GO TO 1000
  925 WRITE(6,930)
  930 FORMAT(6X,63H OVERFLOW OF INTERFACE DEFINITION TABLE - PROCESSING
     $TERMINATED)
      GO TO 1000
  935 WRITE(6,940)
  940 FORMAT(6X,32H TOO MANY EQUIVALENCED VARIABLES)
      GO TO 1000
  945 WRITE(6,950)
  950 FORMAT(6X,61H TOO MANY VARIABLES IN THIS STATEMENT - PROCESSING TE
     $RMINATED)
 1000 WRITE(6,1)
      RETURN
      END
      SUBROUTINE EXPR
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER FNCLOC,OPRA(6),BITPUT,BITGET
      INTEGER D,ASTRIK,EQUALS,STR,A,RPAR
      DATA OPRA/1H+.1H-,1H/,1H(,1H),1H,/
      DATA ASTRIK/1H*/,EQUALS/1H=/,RPAR/1H)/,LPAR/1H(/
      LP=0
      NFUNC=0
      K=0
      IEXPST=NBLOCK+1
      MARGS=0
  200 K=K+1
      CALL GNLE
      IF(JTYP .EQ. 0) RETURN
      IF(JTYP .NE. 1) GO TO 20
      IF(LTYP .EQ. 9 .OR. ITYP .EQ. 6) GO TO 2
      IF(ITYP .EQ. 1 .OR. ITYP .EQ. 35) GO TO 1
      GO TO 5
    1 IF(D(1) .EQ. EQUALS) RETURN
      GO TO 5
    2 IF(D(1) .EQ. RPAR .AND. LP .EQ. 1) RETURN
    5 DO 10 I=1,6
      IF(D(1) .NE. OPRA(I)) GO TO 10
      STR(K)=-I
      IF(I .EQ. 4) GO TO 6
      IF(I .EQ. 5) GO TO 7
      GO TO 100
    6 LP=LP+1
      GO TO 100
    7 LP=LP-1
      GO TO 100
   10 CONTINUE
      IF(D(1) .NE. EQUALS) GO TO 12
      STR(K)=-18
      GO TO 100
   12 IF(D(1) .NE. ASTRIK) GO TO 110
      IF(D(2) .EQ. ASTRIK .AND. M .GT. 1) GO TO 15
      STR(K)=-7
      GO TO 100
   15 STR(K)=-8
      GO TO 100
   20 IF(JTYP .NE. 7) GO TO 30
      IF(LOGID .GT. 9) GO TO 25
      STR(K)=-(LOGID+8)
      GO TO 100
   25 STR(K)=LSTART+440000+M*1000000
      GO TO 100
   30 IF(JTYP .NE. 4) GO TO 40
      IF(IDES .EQ. 0) GO TO 35
      STR(K)=LSTART+420000+M*1000000                                    00000560
      GO TO 100                                                         00000570
   35 CONTINUE                                                          00000580
      STR(K)=LSTART+400000+M*1000000                                    00000590
      GO TO 100                                                         00000600
   40 IF(JTYP .NE. 6) GO TO 50                                          00000610
      STR(K)=LSTART+410000+M*1000000                                    00000620
      GO TO 100                                                         00000630
   50 IF(JTYP .NE. 5) GO TO 55                                          00000640
      STR(K)=LSTART+430000+M*1000000                                    00000650
      GO TO 100                                                         00000660
   55 IF(JTYP .NE. 3) GO TO 60                                          00000662
      STR(K)=LSTART+450000+M*1000000                                    00000664
      IF(ITYP .NE. 8) CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)                00000666
      GO TO 100                                                         00000668
   60 IF(JTYP .NE. 2) GO TO 110                                         00000670
      CALL SEARCH                                                       00000680
      IBETA=0                                                           00000690
      IF(NEXT(JPTR) .NE. LPAR) GO TO 64                                 00000700
      IF(ISRCH(1) .EQ. 0) GO TO 62                                      00000710
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) GO TO 67                      00000720
      CALL SWITCH                                                       00000730
      IBETA=5                                                           00000740
      GO TO 63                                                          00000750
   62 IBETA=5                                                           00000760
      IF(ISRCH(2) .EQ. 1) GO TO 63                                      00000770
      IDTYP=2                                                           00000780
      CALL STORE                                                        00000790
      LOC=NID                                                           00000800
      DO 70 I=1,NLIST                                                   00000810
      IF(ISUBLT(1,I) .NE. IDTBL(1,LOC)) GO TO 70                        00000820
      IF(ISUBLT(2,I) .NE. IDTBL(2,LOC)) GO TO 70                        00000830
      IF(BITGET(ISUBLT(3,I),10,4) .NE. 4) GO TO 63                      00000840
      ITP=BITGET(ISUBLT(3,I),13,3)                                      00000850
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)                            00000860
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ITP,10)                          00000870
      GO TO 63                                                          00000880
   70 CONTINUE                                                          00000890
   63 NFUNC=NFUNC+1                                                     00000900
      IF(NFUNC .GT. 5) GO TO 120                                        00000905
      FNCLOC(NFUNC)=LOC                                                 00000910
      GO TO 68                                                          00000920
   64 IF(ISRCH(2) .NE. 1) GO TO 65                                      00000930
      IF(LOC .NE. IFNCNM)CALL ERROR(10,NXTID(1),NXTID(2),IDM1,IDM2)     00000940
   65 IF(ISRCH(1) .EQ. 1) GO TO 67                                      00000950
      IDTYP=1                                                           00000960
      CALL STORE                                                        00000970
      LOC=NID                                                           00000980
      GO TO 68                                                          00000990
   67 IBETA=BITGET(IDTBL(3,LOC),7,6)                                    00001000
      IF(LOC .NE. IFNCNM) GO TO 68                                      00001010
      IBETA=0                                                           00001020
      LOC=IDES                                                          00001030
   68 CALL IMPTYP                                                       00001040
      IALPH=BITGET(IDTBL(3,LOC),10,3)-1                                 00001050
      JPTR=JPTR-1
      STR(K)=LOC+10000*IALPH+100000*IBETA+1000000*M
  100 NSTR=K
      IF(NSTR .GT. 500) GO TO 130
      GO TO 200
  110 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)
      RETURN
  120 CALL ERROR(90,IDM1,IDM2,IDM3,IDM4)
      STOP
  130 CALL ERROR(91,IDM1,IDM2,IDM3,IDM4)
      STOP
      END
```

```
      SUBROUTINE EXPRCK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP
      DIMENSION IA(5,5)
      INTEGER RHSTYP
      DATA IA/1,0,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,1,1,0,0,0,0,0,1/
      IF(IA(LHSTYP,RHSTYP+1) .EQ. 0) CALL ERROR(27,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE FLOWCK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/FLOW/IFL,IRP
      DIMENSION IPATH(100),ISTCK(100)
      INTEGER FLWLST(100),BRANCH,STATRA
      INTEGER BITPUT,BITGET
      EQUIVALENCE (IPATH(1),A(1)),(ISTCK(1),A(101)),(FLWLST(1),A(201))
      IF(IFL .EQ. -1) GO TO 3000
      IRX=0
      NSTCK=0
      NFLOW=0
      NOC=0
      NPTHS=0
      IBLKST=1
      CALL CHKLST
      WRITE(6,8)
    8 FORMAT(1H1,38H****** RESULTS OF FLOW ANALYSIS ******//)
    5 DO 10 I=1,NID
   10 IDTBL(4,I)=IDTBL(8,I)
   12 IF(NFLOW .EQ. 0) GO TO 20
      NOC=0
      DO 15 I=1,NFLOW
      IF(IABS(FLWLST(I)) .NE. IBLKST) GO TO 15
      NOC=NOC+1
   15 CONTINUE
      IF(NOC .GT. IRP) GO TO 1500
   20 NFLOW=NFLOW+1
      IF(NFLOW .GT. 100) GO TO 4000
      FLWLST(NFLOW)=IBLKST
      IEND=BITGET(IBLOCK(IBLKST),24,12)-1
      IF(IEND .EQ. -1) IEND=NBLOCK
      NBR=BITGET(IBLOCK(IBLKST),6,6)
      ISTART=IEND-NBR+1
      IBLKST=NXTBLK(ISTART,IEND)
      IF(NBR .EQ. 1) GO TO 25
      FLWLST(NFLOW)=-FLWLST(NFLOW)
      NSTCK=NSTCK+1
      IF(NSTCK .GT. 100) GO TO 5000
      ISTCK(NSTCK)=-NXTBLK(IEND,IEND)
      IF(NBR .EQ. 2) GO TO 25
      DO 22 J=3,NBR
      NSTCK=NSTCK+1
      IF(NSTCK .GT. 100) GO TO 5000
   22 ISTCK(NSTCK)=NXTBLK(IEND-J+2,IEND)
   25 IF(IBLKST .NE. 0) GO TO 12
      NPATH=0
      NPTHS=NPTHS+1
      DO 1000 I=1,NFLOW
      BRANCH=IABS(FLWLST(I))
      ISTART=BRANCH+1
      NXBLOK=BITGET(IBLOCK(BRANCH),24,12)
      IF(NXBLOK .EQ. 0) NXBLOK=NBLOCK+1
      ISL=BITGET(IBLOCK(BRANCH),32,8)
      ILOOP=BITGET(IBLOCK(BRANCH),12,6)
      NBR=BITGET(IBLOCK(BRANCH),6,6)
      IEND=NXBLOK-NBR-1
      IF(ISL .EQ. 0) GO TO 45
      NPATH=NPATH+1
      IPATH(NPATH)=STATRA(1,ISL)
      STATRA(2,ISL)=BITPUT(STATRA(2,ISL),1,18)
   45 IF(IBLOCK(ISTART) .LT. 1000) GO TO 1000
      DO 500 J=ISTART,IEND
      IT=IBLOCK(J)/1000
      LOC=IBLOCK(J)-IT*1000
      GO TO(50,60,70,80,90,100,200),IT
   50 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120
      IF(IDTBL(4,LOC) .EQ. 2) GO TO 55
      IF(IDTBL(4,LOC) .EQ. 4) GO TO 180
   52 IDTBL(4,LOC)=1
   30 IF(BITGET(IDTBL(3,LOC),17,1) .EQ. 0) GO TO 500
      KTYPE=BITGET(IDTBL(3,LOC),10,3)
      NXQV=LOC
   53 NXQV=IDTBL(10,NXQV)
      IF(NXQV .EQ. LOC) GO TO 500
      IF(BITGET(IDTBL(3,NXQV),10,3) .EQ. KTYPE) GO TO 54
      IDTBL(4,NXQV)=0
      GO TO 53
   54 IDTBL(4,NXQV)=1
      GO TO 53
   55 IF(ILOOP .EQ. 0) GO TO 52
   57 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 52
      ILOOP=ISTACK(3,ILOOP)
      GO TO 57
   60 IF(IDTBL(4,LOC) .EQ. 0) GO TO 140
      IF(TDTBL(4,LOC) .EQ. 3) GO TO 130
      GO TO 500
   70 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120
      IF(IDTBL(4,LOC) .EQ. 2) GO TO 75
   72 IDTBL(4,LOC)=2
      GO TO 500
   75 IF(ILOOP .EQ. 0) GO TO 72
   77 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 72
      ILOOP=ISTACK(3,ILOOP)
      GO TO 77
   80 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120
      IF(IDTBL(4,LOC) .EQ. 2) GO TO 85
   82 IDTBL(4,LOC)=3
      GO TO 500
   85 IF(ILOOP .EQ. 0) GO TO 82
   87 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 82
      ILOOP=ISTACK(3,ILOOP)
      GO TO 87
   90 IF(IDTBL(4,LOC) .NE. 3) GO TO 150
      GO TO 500
  100 IDTBL(4,LOC)=0
      GO TO 500
  180 JLOOP=BITGET(IDTBL(3,LOC),32,9)
  185 IF(ILOOP .EQ. JLOOP) GO TO 160
      ILOOP=ISTACK(3,ILOOP)
      IF(ILOOP .EQ. 0) GO TO 30
      GO TO 185
  200 IF(IDTBL(4,LOC) .EQ. 0) GO TO 140
      IF(IDTBL(4,LOC) .EQ. 3) GO TO 130
      IDTBL(4,LOC)=4
      GO TO 500
  110 IERC=67
      GO TO 400
  120 IERC=68
      GO TO 400
  130 IERC=69
      GO TO 400
  140 IERC=70
      GO TO 400
  150 IERC=71
      GO TO 400
  160 IERC=72
  400 IRX=1
      IF(BITGET(IDTHL(3,LOC),20,1) .EQ. 1) GO TO 500
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,20)
      CALL ERROR(IERC,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
      IF(IERC .NE. 70 .OR. NPATH .EQ. 0) GO TO 500
      WRITE(6,410) (IPATH(K),K=1,NPATH)
  410 FORMAT(6X,15H ALONG THE PATH,(10I6))
  500 CONTINUE
 1000 CONTINUE
 1500 IF(FLWLST(NFLOW) .GT. 0) GO TO 1600
      IBLKST=IABS(ISTCK(NSTCK))
      IF(ISTCK(NSTCK) .LT. 0) FLWLST(NFLOW)=-FLWLST(NFLOW)
      NSTCK=NSTCK-1
      NOC=0
      GO TO 5
 1600 NFLOW=NFLOW-1
      IF(NFLOW .GT. 0) GO TO 1500
      IF(NLABEL .EQ. 0) GO TO 2010
      DO 2000 J=1,NLABEL
      IF(BITGET(STATRA(2,J),6,6) .EQ. 28) GO TO 2000
      IF(BITGET(STATRA(2,J),18,3) .EQ. 1) GO TO 2000
      WRITE(6,1800) STATRA(1,J)
      IRX=1
 1800 FORMAT(6X,57H THERE IS NO COMPLETE PATH THAT CONTAINS STATEMENT NU
     1MBER,I6)
 2000 CONTINUE
 2010 IF(IRX .EQ. 0) WRITE(6,2020)
 2020 FORMAT(//6X,16H NO ERRORS FOUND)
      WRITE(6,2100) NPTHS
 2100 FORMAT(//////6X,25H NUMBER OF PATHS CHECKED-,I6)
      RETURN
 3000 WRITE(6,3001)
 3001 FORMAT(//31X,57H FLOW ANALYSIS WAS NOT PERFORMED DUE TO ERRORS IN
     *PROGRAM)
      IFL=IRP+1
      RETURN
 4000 WRITE(6,4001)
 4001 FORMAT(//29X,63H TABLE OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS
     $IS TERMINATED)
      RETURN
 5000 WRITE(6,5001)
 5001 FORMAT(//29X,63H STACK OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS
     $IS TERMINATED)
      RETURN
      END
      SUBROUTINE FNCSTR
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER FNCLOC,BITPUT,BITGET
      IF(NFUNC .EQ. 0) RETURN
      DO 40 I=1,NFUNC
      LOC=FNCLOC(I)
      IF(BITGET(IDTBL(3,LOC),19,1) .EQ. 1) GO TO 50
      NARG=IFNCRA(I,1)
      IVAR=0
      ITP=BITGET(IDTBL(3,LOC),10,3)
      IF(BITGET(IDTBL(3,LOC),18,1) .EQ. 1) GO TO 20
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)
      DO 5 J=1,NLIST
      IF(IDTBL(1,LOC) .NE. ISUBLT(1,J)) GO TO 5
      IF(IDTBL(2,LOC) .NE. ISUBLT(2,J)) GO TO 5
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),J,32)
      LISTLC=J
      GO TO 21
    5 CONTINUE
      CALL ERROR(52,IDM1,IDM2,IDM3,IDM4)
      NLIST=NLIST+1
      IF(NLIST .GT. 200) GO TO 60
      ISUBLT(1,NLIST)=IDTBL(1,LOC)
      ISUBLT(2,NLIST)=IDTBL(2,LOC)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,32)
      IPTR=NINTFC+1
      IF(ITYP .EQ. 8 .AND. I .EQ. 1) ITP=0
      ISUBLT(3,NLIST)=BITPUT(0,ITP,13)
      ISUBLT(3,NLIST)=BITPUT(ISUBLT(3,NLIST),NARG,6)
      ISUBLT(4,NLIST)=IPTR
      NINTFC=IPTR+(NARG-1)/3
      IF(NINTFC .GT. 600) GO TO 70
      DO 10 J=IPTR,NINTFC
   10 INTFAC(J)=IFNCRA(I,J-IPTR+2)
      GO TO 40
   20 LISTLC=BITGET(IDTBL(3,LOC),32,9)
   21 IPTR=ISUBLT(4,LISTLC)
      IF(BITGET(ISUBLT(3,LISTLC),14,1) .EQ. 1) GO TO 22
      NAR2=BITGET(ISUBLT(3,LISTLC),6,6)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),I7,I8)
      NARGS=MIN0(NARG,NAR2)
      GO TO 24
   22 IVAR=1
      IF(NARG .LT. 2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),I7,I8)
      NARGS=NARG
      ITP1=BITGET(INTFAC(IPTR),3,3)
      NDIM1=BITGET(INTFAC(IPTR),6,3)
   24 IF(ITYP .EQ. 8 .AND. I .EQ. 1) GO TO 25
      IF(BITGET(ISUBLT(3,LISTLC),10,4) .EQ. 4) GO TO 25
      JTP=BITGET(ISUBLT(3,LISTLC),13,3)
      IF(JTP .NE. ITP) CALL ERROR(49,IDTBL(1,LOC),IDTBL(2,LOC),I7,I8)    00000540
   25 NDPTR=IPTR+(NARGS-1)/3                                              00000550
      KOUNT=0                                                             00000560
      DO 32 K=IPTR,NDPTR                                                  00000570
      ICOL1=-6                                                            00000580
      ICOL2=-3                                                            00000590
      DO 32 J=1,3                                                         00000600
      KOUNT=KOUNT+1                                                       00000610
      IF(KOUNT .GT. NARGS) GO TO 40                                       00000620
      ICOL1=ICOL1+9                                                       00000630
      ICOL2=ICOL2+9                                                       00000640
      IF(IVAR .EQ. 1) GO TO 26                                            00000650
      ITP1=BITGET(INTFAC(K),ICOL1,3)                                      00000660
      NDIM1=BITGET(INTFAC(K),ICOL2,3)                                     00000670
   26 ITP2=BITGET(1FNCRA(I,K-IPTR+2),ICOL1,3)                             00000680
      NDIM2=BITGET(IFNCRA(I,K-IPTR+2),ICOL2,3)                            00000690
      IF(NDIM1 .NE. NDIM2) CALL ERROR(50,KOUNT,I7,I8,I9)                  00000700
      IF(ITP2 .EQ. 0) GO TO 32                                            00000710
      IF(ITP1 .EQ. 0) GO TO 28                                            00000720
      IF(ITP1 .NE. ITP2) CALL ERROR(51,KOUNT,I7,I8,I9)                    00000730
      GO TO 32                                                            00000740
   28 INTFAC(K)=BITPUT(INTFAC(K),ITP2,ICOL1)                              00000750
   32 CONTINUE                                                            00000760
      GO TO 40                                                            00000770
   50 CALL STFNC(I)                                                       00000780
   40 CONTINUE                                                            00000790
      RETURN                                                              00000800
   60 CALL ERROR(92,IDM1,IDM2,IDM3,IDM4)                                  00000802
      STOP                                                                00000804
   70 CALL ERROR(93,IDM1,IDM2,IDM3,IDM4)                                  00000806
      STOP                                                                00000808
      END                                                                 00000810
      SUBROUTINE FORM
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      COMMON/VAR/VFOR,NCHAR,NCHARP,CHAR,NDICT
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON /STRING/ NTYPE,NSTR,STR
      INTEGER BITPUT,BITGET
      INTEGER VFOR(30),CHAR,STR(1)
      LOGICAL ERRFLG
      DATA IX/1HX/
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      IF(CHAR .NE. IX) NDICT=-NDICT
      NCHARP=NCHARP+1
      STR(NCHARP)=NDICT
      IF(.NOT. ERRFLG) RETURN
      NCHAR=NCHAR+1
      NC=1+(NCHARP-1)/4
      ICHAR=BITGET(CHAR,8,8)
      VFOR(NC)=BITPUT(VFOR(NC),ICHAR,8*NCHAR)
      IF(NCHAR .EQ. 4) NCHAR=0
C     COMPLETE
      RETURN
25000 CONTINUE
      GO TO 25001
      END
      SUBROUTINE FORMEL
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER B(50),D
      DIMENSION IC(10)
      DATA IC/75,77,78,91,92,93,96,97,107,126/
      DATA IH/1HH/
      GO TO (100,10,20,100,40,100,100,100),JTYP
   10 CALL CAA(D,M,NXTID)
      RETURN
   20 DO 25 I=1,10
      IF(D(I) .NE. IH) GO TO 25
      CALL CAI(D,I-1,N2)
      IF(N2 .LT. 1) GO TO 110
      M=N2+I
      IF(M .GT. 500) CALL ERROR(5,IDM1,IDM2,IDM3,IDM4)
      JPTR=LSTART+M
      IST=I+1
      DO 22 J=IST,M
      ICHAR=BITGET(D(J),8,8)
      DO 23 II=1,10
      IF(ICHAR .EQ. IC(II)) GO TO 120
   23 CONTINUE
   22 CONTINUE
      IF(ITYP .EQ. 28) RETURN
      IF(N2 .GT. 4) CALL ERROR(5,IDM1,IDM2,IDM3,IDM4)
      RETURN
   25 CONTINUE
      CALL ERROR(3,IDM1,IDM2,IDM3,IDM4)
      RETURN
   40 CALL CAI(D,M,N2)
  100 RETURN
  110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
  120 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE FRMAT
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP
      DIMENSION RPLOC(20),IALPH(6)
      INTEGER A,RPLOC,AICH,RPAR,BLANK,SEPST,SEPND,DIR
      DATA BLANK/1H /,AICH/1HH/,LPAR/1H(/,RPAR/1H)/
      DATA IALPH/1HF,1HO,1HR,1HM,1HA,1HT/
      IFRMT=0
      DO 4 I=1,6
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 70
    4 CONTINUE
      NSTART=JPTR
      IF(NEXT(JPTR) .NE. LPAR) GO TO 70
      DO 10 I=1,N
      IF(ITYPE(JPTR) .EQ. 2) GO TO 1
      IF(JPTR .GT. N) GO TO 12
      GO TO 10
    1 JPTR=JPTR-1
      CALL GNLE
      IF(JTYP .NE. 3) GO TO 10
      J1=JPTR-1
      IH=0
      DO 5 J=LSTART,J1
      IF(IH .EQ. 1) GO TO 3
      IF(A(J) .EQ. AICH) IH=1
      GO TO 5
    3 A(J)=BLANK
    5 CONTINUE
   10 CONTINUE
   12 NPAR=0
      NRP=0
      DO 20 I=NSTART,N
      IF(A(I) .NE. LPAR) GO TO 15
      NPAR=NPAR+1
      IF(NPAR .GT. 3) GO TO 70
      GO TO 20
   15 IF(A(I) .NE. RPAR) GO TO 20
      NPAR=NPAR-1
      NRP=NRP+1
      RPLOC(NRP)=I
      IF(NPAR .LT. 0) GO TO 70
   20 CONTINUE
      IF(NPAR .NE. 0) GO TO 70
      JPTR=RPLOC(NRP)+1
      IF(NEXT(JPTR) .NE. BLANK) GO TO 70
      DO 60 I=1,NRP
      IGPND=RPLOC(I)
      DO 25 J=1,N
      K=IGPND-J
      IF(A(K) .NE. LPAR) GO TO 25
      IGPST=K
      GO TO 30
```

```
25 CONTINUE
30 CALL GROUP
   IF(IGRP .EQ. 0) RETURN
   IF(I .EQ. NRP) GO TO 65
   JPTR=IGPST-1
31 IPV=JPTR
   IF(IPREV(IPV) .EQ. 2) GO TO 31
   IGPST=JPTR+2
35 SEPST=IGPND+1
   DIR=1
   CALL SEPAR
   IF(ISEP .NE. 1) GO TO 40
   IGPND=SEPND
   GO TO 50
40 JPTR=SEPST
   IF(NEXT(JPTR) .NE. RPAR) GO TO 70
   SEPST=IGPST-1
   DIR=-1
   CALL SEPAR
   IF(ISEP .NE. 1) GO TO 45
   IGPST=SEPND
   GO TO 50
45 IF(A(SEPND) .NE. LPAR) GO TO 70
50 DO 55 J=IGPST,IGPND
   A(J)=BLANK
55 CONTINUE
60 CONTINUE
65 IFRMT=1
   RETURN
70 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
   RETURN
   END
```

```
      SUBROUTINE GENROL
      COMMON A(1326).D(500).IDTBL(11.500).INITID(3).LASTID(3).ISRCH(3),
     * NXTID(2).JPTR.N.M.JTYP.LSTART.N2.IFNCNM.LOGID.IDTYP.NID.
     * LOC.LTYP.ITYP.IBLKDT.MODE.IERR.IDES
      COMMON/GLOBAL/NBLK.NREF.NSUBS.BLKTBL(200).EXTTBL(100).ISUBS(100)
      COMMON/LIST/NLIST.NINTFC.ISUBLT(4.200).INTFAC(600)
      COMMON/INPOUT/NCALL.IN.IOP
      COMMON/WASTE/IDUM(63)
      DIMENSION JSES(2)
      INTEGER BLKTBL.EXTTBL.BITGET
      DATA JSES(1).JSES(2)/4HSESC.2HOM/
      DATA ID/2HD./.IJ/2HJ)/
      WRITE(6.2)
    2 FORMAT(1H1)
      IF(NBLK .EQ. 0) GO TO 6
      K=-1
      DO 5 I=1.NBLK
      K=K+1
      INDEX=BLKTBL(I)
      ISZ=BITGET(ISUBLT(3.INDEX).32.15)
      WRITE(IOP.3) ISUBLT(1.INDEX).ISUBLT(2.INDEX).K.ISZ
    3 FORMAT(5X.8H COMMON/.A4.A2.3H/IX.I2.1H(.I6.1H))
    5 CONTINUE
    6 WRITE(IOP.7) MODE
    7 FORMAT(5X.4H J=1/5X.6H MODE=.I1/5X.13H DO 10 I=1.13/5X.6H J=J-1)
      IF(NBLK .EQ. 0) GO TO 22
      K=-1
      DO 20 I=1.NBLK
      K=K+1
      KK=1000+K
      INDEX=BLKTBL(I)
      ISZ=BITGET(ISUBLT(3.INDEX).32.15)
      WRITE(IOP.10) KK.ISZ,K,KK
   10 FORMAT(5X.4H DO .I4,5H K=1..I6/5X.3H IX.I2.5H(K)=1/1X.I4.
     $ 9H CONTINUE)
      IF(ICOMP(JSES.ISUBLT.INDEX.4) .EQ. 0) GO TO 20
      WRITE(IOP.15) K.K,K
   15 FORMAT(5X.3H IX,I2.7H(17)=10/5X.3H IX,I2.7H(20)=11/
     * 5X,3H IX,I2.7H(23)=12)
   20 CONTINUE
   22 NARG=BITGET(IDTBL(3.1).7.6)
      DO 30 I=1.NARG
      IF(I .EQ. NARG) GO TO 25
      IDUM(I)=ID
      GO TO 30
   25 IDUM(I)=IJ
   30 CONTINUE
      WRITE(IOP.35) IDTBL(1.1).IDTBL(2.1).(IDUM(I).I=1.NARG)
   35 FORMAT(5X.6H CALL .A4.A2.1H(.41A2/5X.1H1.1X.22A2)
      WRITE(IOP.40)
   40 FORMAT(5X.24H IF(MODE .EQ. 3) GO TO 5/5X.14H CALL MODID(J)/
     $ 3X.12H 5 ENDFILE 3/2X.12H 10 CONTINUE/5X.12H CALL CMPARE/
     $ 5X.10H REWIND 13/5X.10H REWIND 14/5X.10H REWIND 15/5X.5H STOP/
     $ 5X,4H END)
      RETURN
      END
      SUBROUTINE GLOTAB
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      INTEGER BLKTBL,BITGET,EXTTBL,KLAS(5,7)
      DATA KLAS/4HUSER,4H SUP,4HPLIE,1HD,1H ,4HSUBR,4HOUTI,4HNE M,
     $ 4HODUL,1HE,4HFUNC,4HTION,4H MOD,3HULE,1H ,4HANCI,4HLLAR,4HY SU,
     $ 4HBPRO,4HGRAM,4HANSI,4H FUN,4HCTIO,1HN,1H ,4HMAIN,4H PRO,4HGRAM,
     $ 1H ,1H ,4HEXTR,4HAORD,4HINAR,4HY SU,4HBPR./
      DATA IBLNK/1H /
      WRITE(6,1)
    1 FORMAT(1H1,48X,23H GLOBAL REFERENCE TABLE)
      IF(NREF .EQ. 0) GO TO 25
      WRITE(6,2)
    2 FORMAT(//50X,20H EXTERNAL REFERENCES)
      DO 20 I=1,NREF
      INDEX=EXTTBL(I)
      J=BITGET(ISUBLT(3,INDEX),10,4)
      WRITE(6,10) ISUBLT(1,INDEX),ISUBLT(2,INDEX),(KLAS(II,J+1),II=1,5)
   10 FORMAT(45X,A4,A2,4X,5A4)
   20 CONTINUE
   25 IF(NBLK .EQ. 0 .OR. (NBLK .EQ. 1 .AND. ISUBLT(1,BLKTBL(1)) .EQ.
     $ IBLNK)) GO TO 40
      WRITE(6,30)
   30 FORMAT(//49X,23H LABELLED COMMON BLOCKS/43X,11H BLOCK NAME,7X,
     $ 5H SIZE,7X,6H CLASS)
      DO 38 J=1,NBLK
      INDEX=BLKTBL(J)
      ICAT=BITGET(ISUBLT(3,INDEX),10,4)-6
      ISZ=BITGET(ISUBLT(3,INDEX),32,15)
   35 FORMAT(46X,A4,A2,5X,I8,5X,9HCATEGORY ,I2)
      WRITE(6,35) ISUBLT(1,INDEX),ISUBLT(2,INDEX),ISZ,ICAT
   38 CONTINUE
   40 WRITE(6,45)
   45 FORMAT(///48X,24H SUBROUTINES ENCOUNTERED)
      DO 60 I=1,NSUBS
      INDEX=ISUBS(I)
      J=BITGET(ISUBLT(3,INDEX),10,4)
      WRITE(6,10) ISUBLT(1,INDEX),ISUBLT(2,INDEX),(KLAS(II,J+1),II=1,5)
   60 CONTINUE
      RETURN
      END
      SUBROUTINE GNLE
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LOGIC/LOG,LOGST
      COMMON/REALNO/IREAL,IRELND,IP
      INTEGER A,D,BLANK,PLUS,EQUALS,SLASH,RPAR,COMMA,ASTRIK,
     1 AICH,DECPT
      DATA BLANK/1H /,PLUS/1H+/,MINUS/1H-/,EQUALS/1H=/,SLASH/1H//,
     * RPAR/1H)/,COMMA/1H,/,ASTRIK/1H*/,AICH/1HH/,LPAR/1H(/,DECPT/1H./
      JTYP=0
      NXT=NEXT(JPTR)
      IF(NXT .EQ. BLANK) RETURN
      LSTART=JPTR-1
      IF(NXT .EQ. PLUS) GO TO 1
      IF(NXT .EQ. RPAR) GO TO 1
      IF(NXT .EQ. MINUS) GO TO 1
      IF(NXT .EQ. SLASH) GO TO 1
      IF(NXT .EQ. COMMA) GO TO 1
      IF(NXT .EQ. EQUALS) GO TO 1
      GO TO 2
    1 JTYP=1
      M=1
      GO TO 90
    2 IF(NXT .NE. ASTRIK) GO TO 4
      IF(NEXT(JPTR) .NE. ASTRIK) GO TO 1
      M=2
      JTYP=1
      GO TO 90
    4 IF(NXT .NE. LPAR) GO TO 40
      IF(LSTART .EQ. 1) GO TO 10
      IM1=LSTART-1
      IF(IPREV(IM1) .NE. 3) GO TO 1
   10 JPTR=LSTART+1
      NXT=NEXT(JPTR)
      IF(NXT .EQ. BLANK) GO TO 120
      IF(NXT .NE. PLUS .AND. NXT .NE. MINUS) GO TO 22
      IP=JPTR
      GO TO 24
   22 IP=JPTR-1
   24 CALL REALCK
      IF(IREAL .EQ. 0) GO TO 1
      IF(IDES .EQ. 1) GO TO 1
      JPTR=IRELND+1
      IF(NEXT(JPTR) .NE. COMMA) GO TO 1
      NXT=NEXT(JPTR)
      IF( NXT .NE. PLUS .AND. NXT .NE. MINUS) GO TO 30
      IP=JPTR
      GO TO 35
   30 IP=JPTR-1
   35 CALL REALCK
      IF(IREAL .EQ. 0) GO TO 120
      IF(IDES .EQ. 1) GO TO 120
      JPTR=IRELND+1
      IF(NEXT(JPTR) .NE. RPAR) GO TO 120
      JTYP=6
      M=JPTR-LSTART
      GO TO 90
   40 IF(NXT .NE. DECPT) GO TO 50
      ITP=ITYPE(JPTR)
      GO TO (42,44,120),ITP
   42 LOGST=LSTART+1
      CALL LOGCHK
      IF(LOG .EQ. 0) GO TO 120
      JTYP=7
      M=JPTR-LSTART
      GO TO 90
   44 IP=LSTART
      CALL REALCK
      IF(IREAL .EQ. 0) GO TO 120
      JTYP=4
      M=IRELND-LSTART+1
      GO TO 90
   50 JPTR=LSTART
      IF(ITYPE(JPTR) .NE. 2) GO TO 85
      IF(ITYP .EQ. 28) GO TO 54
      IP=LSTART
      CALL REALCK
      IF(IREAL .EQ. 0) GO TO 54
      JTYP=4
      M=IRELND-LSTART+1
      GO TO 90
   54 JPTR=LSTART+1
   55 IF(ITYPE(JPTR) .EQ. 2) GO TO 57
      GO TO 65
   57 IF(JPTR .GT. N) GO TO 60
      GO TO 55
   60 M=N-LSTART+1
      JTYP=5
      GO TO 90
   65 IF(A(JPTR-1) .NE. AICH) GO TO 67
      IF(ITYP .EQ. 8 .OR. ITYP .EQ. 28 .OR. ITYP .EQ. 27) GO TO 70
   67 M=JPTR-LSTART-1
      JTYP=5
      GO TO 90
   70 IF(JPTR .GT. N) GO TO 120
      M=N-LSTART+1
      IF(M .GT. 500) M=500
      JTYP=3
      GO TO 90
   85 JPTR=LSTART
      IF(ITYPE(JPTR) .NE. 1) GO TO 120
      IF(ITYP .EQ. 28) GO TO 100
   88 CONTINUE
      IF(ITYPE(JPTR) .NE. 3) GO TO 86
      M=JPTR-LSTART-1
      JTYP=2
      GO TO 90
   86 IF(JPTR .GT. N) GO TO 87
      GO TO 88
```

```
100 M=1
    JTYP=2
    GO TO 90
 87 M=N-LSTART+1
    JTYP=2
 90 CONTINUE
    DO 91 L=1,M
    LL=LSTART+L-1
    D(L)=A(LL)
 91 CONTINUE
    JPTR=LSTART+M
    CALL SQUEEZ
    CALL FORMEL
    RETURN
120 CONTINUE
    JTYP=8
    RETURN
    END
```

```
    SUBROUTINE GOTO
    COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
   * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
   * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
    COMMON/LABELS/STATRA(2,200),NLABEL
    COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
    DIMENSION IALPH(4)
    INTEGER STATRA,BLANK
    INTEGER BITPUT
    DATA IALPH/1HG,1HO,1HT,1HO/
    DATA BLANK/1H /
    DO 5 I=1,4
    IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 10
  5 CONTINUE
    CALL GNLE
    IF(JTYP .NE. 5) GO TO 10
    CALL STSRCH
    STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
    IF(NEXT(JPTR) .NE. BLANK) GO TO 10
    NBLOCK=NBLOCK+1
    IBLOCK(NBLOCK)=LOC
    NBRNCH=1
    NB=1
    RETURN
 10 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
    RETURN
    END
```

```
      SUBROUTINE GROUP
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP
      INTEGER A,RPAR,SEPST,SEPND,DIR
      DATA RPAR/1H)/
      JPTR=IGPST+1
      IF(NEXT(JPTR) .EQ. RPAR) GO TO 20
      SEPST=JPTR-1
      DIR=1
      CALL SEPAR
      IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30
      IF(ISEP .EQ. 0) IDESST=SEPST
      IF(ISEP .EQ. 1) IDESST=SEPND+1
      JPTR=IDESST
      IF(NEXT(JPTR) .EQ. RPAR) GO TO 20
      SEPST=IGPND-1
      DIR=-1
      CALL SEPAR
      IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30
      DIR=1
   10 CONTINUE
      CALL DESCRP
      IF(IDES .EQ. 0) GO TO 40
      SEPST=IDESND+1
      JPTR=SEPST
      IF(NEXT(JPTR) .EQ. RPAR) GO TO 20
      CALL SEPAR
      IF(ISEP .EQ. 0 .OR. ISEP .EQ. -1) GO TO 30
      IDESST=SEPND+1
      JPTR=IDESST
      IF(NEXT(JPTR) .NE. RPAR) GO TO 10
   20 IGRP=1
      RETURN
   30 IGRP=0
      CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
   40 CALL ERROR(88,IDESST,IDM2,IDM3,IDM4)
      IGRP=0
      RETURN
      END
      SUBROUTINE GRT
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      INTEGER EXTTBL,BLKTBL,BLANK,BITPUT,BITGET
      DATA BLANK/1H /
      WRITE(6,1)
    1 FORMAT(//)
      ISUB=INITID(2)
      IF(ISUB .EQ. 0) GO TO 15
   10 ISUB=IDTBL(4,ISUB)
      IF(ISUB .EQ. 0) GO TO 15
      IF(BITGET(IDTBL(3,ISUB),19,1) .EQ. 1) GO TO 10
      IF(NREF .EQ. 0) GO TO 4
      DO 3 K=1,NREF
      INDEX=EXTTBL(K)
      IF(IDTBL(1,ISUB) .EQ. ISUBLT(1,INDEX) .AND. IDTBL(2,ISUB) .EQ.
     $ ISUBLT(2,INDEX)) GO TO 10
    3 CONTINUE
    4 NREF=NREF+1
      IF(NREF .GT. 100) GO TO 50
      EXTTBL(NREF)=BITGET(IDTBL(3,ISUB),32,9)
      IF(MODE .EQ. 1) GO TO 10
      KLAS=BITGET(ISUBLT(3,EXTTBL(NREF)),10,4)
      IF(KLAS.EQ.1.OR.KLAS.EQ.2)WRITE(9)IDTBL(1,ISUB),IDTBL(2,ISUB)
      GO TO 10
   15 IBLK=INITID(3)
   20 IF(IBLK .EQ. 0) RETURN
      IF(IDTBL(1,IBLK) .EQ. BLANK) GO TO 45
      DO 25 I=1,NLIST
      IF(IDTBL(1,IBLK) .NE. ISUBLT(1,I)) GO TO 25
      IF(IDTBL(2,IBLK) .NE. ISUBLT(2,I)) GO TO 25
      LISTLC=I
      IF(BITGET(ISUBLT(3,I),10,4) .EQ. 7) GO TO 30
      IF(BITGET(ISUBLT(3,I),32,15) .NE. 0) GO TO 30
      ISZ=IDTBL(5,IBLK)
      ISUBLT(3,I)=BITPUT(ISUBLT(3,I),ISZ,32)
      GO TO 30
   25 CONTINUE
      CALL ERROR(62,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,IBLK)
      ISUBLT(2,NLIST)=IDTBL(2,IBLK)
      ISZ=IDTBL(5,IBLK)
      ISUBLT(3,NLIST)=BITPUT(ISZ,10,10)
      LISTLC=NLIST
   30 IDTBL(3,IBLK)=BITPUT(IDTBL(3,IBLK),LISTLC,32)
      IF(NBLK .EQ. 0) GO TO 40
      DO 35 K=1,NBLK
      IF(LISTLC ,EQ. BLKTBL(K)) GO TO 45
   35 CONTINUE
   40 NBLK=NBLK+1
      IF(NBLK .GT. 200) GO TO 60
      BLKTBL(NBLK)=LISTLC
   45 IBLK=IDTBL(4,IBLK)
      GO TO 20
   50 CALL ERROR(59,IDM1,IDM2,IDM3,IDM4)
      STOP
   60 CALL ERROR(60,IDM1,IDM2,IDM3,IDM4)
      STOP
      END
      FUNCTION ICOMP(IVAR,JVAR,NVAR,NROW)
      DIMENSION IVAR(1),JVAR(NROW,1)
      ICOMP=0
      IF(IVAR(1) .NE. JVAR(1,NVAR)) RETURN
      IF(IVAR(2) .NE. JVAR(2,NVAR)) RETURN
      ICOMP=1
      RETURN
      END
```

```
      SUBROUTINE IMPTYP
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH(6)
      INTEGER D,BITPUT,BITGET
      DATA IALPH/1HI,1HJ,1HK,1HL,1HM,1HN/
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 20
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)
      DO 10 I=1,6
      IF(D(1) .NE. IALPH(I)) GO TO 10
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),4,10)
      GO TO 20
   10 CONTINUE
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,10)
   20 IF(ITYP .LE. 18) IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,21)
      RETURN
      END
      SUBROUTINE INIT
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/STFUNC/NSTFNC,ISTFNC(10)
      INTEGER RHSTYP
      INTEGER A,EQUALS,COMMA,RPAR,STR
      INTEGER BITPUT,BITGET,FNCLOC
      DATA EQUALS/1H=/,LPAR/1H(/,COMMA/1H,/,RPAR/1H)/
      IFN=0
      NTYPE=1
      IPTR=JPTR
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 40
      CALL SEARCH
      IF(NEXT(JPTR) .NE. EQUALS) GO TO 6
      LOC2=0
      IF(ISRCH(2) .NE. 1) GO TO 2
      IF(LOC .NE. IFNCNM) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IFN=1
      CALL IMPTYP
      LOC2=LOC
    2 IF(ISRCH(1)'.NE. 1) GO TO 18
      IF(LOC2 .EQ. 0) GO TO 4
      LOC=IDES
      GO TO 5
   18 IDTYP=1
      CALL STORE
      LOC=NID
      IF(LOC2 .EQ. 0) GO TO 4
      IDTBL(3,LOC)=IDTBL(3,LOC2)
      GO TO 5
    4 CALL IMPTYP
    5 LHSTYP=BITGET(IDTBL(3,LOC),10,3)
      IF(LHSTYP .EQ. 5) NTYPE=2
      GO TO 30
    6 IF(A(JPTR-1) .NE. LPAR) GO TO 40
      IF(ISRCH(1) .EQ. 1) GO TO 12
      IF(ISRCH(2) .NE. 1) GO TO 15
      CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      GO TO 8
   15 IDTYP=2
      CALL STORE
      LOC=NID
      GO TO 8
    7 CALL SWITCH
    8 CALL IMPTYP
      LHSTYP=BITGET(IDTBL(3,LOC),10,3)
      IF(LHSTYP .EQ. 5) NTYPE=2
      NARG=0
      ITYP=35
      NSTFNC=NSTFNC+1
      IF(NSTFNC .GT. 10) GO TO 60
      ISTFNC(NSTFNC)=LOC
      LOC1=LOC
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,19)
   10 CALL GNLE
      IF(JTYP .NE. 2) GO TO 50
      NARG=NARG+1
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(54,NARG,IDM2,IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 20
      IDTYP=1
      CALL STORE
      LOC=NID
      GO TO 25
   20 IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(54,NARG,I7,I8,I9)
   25 CALL IMPTYP
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 10
      IF(A(JPTR-1) .NE. RPAR) GO TO 40
      IDTBL(3,LOC1)=BITPUT(IDTBL(3,LOC1),NARG,7)
      IF(NEXT(JPTR) .EQ. EQUALS) GO TO 32
      GO TO 40
   12 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) GO TO 7
      CALL IMPTYP
      LHSTYP=BITGET(IDTBL(3,LOC),10,3)
      IF(LHSTYP .EQ. 5) NTYPE=2
      JPTR=IPTR
      JBLOCK=NBLOCK+1
      CALL EXPR
      CALL PARSE
      CALL BLKSTR
      IBLOCK(JBLOCK)=IBLOCK(JBLOCK) - 1000
      GO TO 32
   30 NBLOCK=NBLOCK+1
      JBLOCK=NBLOCK
      IBLOCK(NBLOCK)=1000+LOC
   32 NTMS=0
      IPTR=JPTR
      CALL EXPR
      IF(JPTR .LE.N) CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      CALL PARSE
      CALL FNCSTR
      CALL EXPRCK
      IF(ITYP .EQ. 35) GO TO 36
      CALL BLKSTR
      IBLOCK(NBLOCK+1)=IBLOCK(JBLOCK)
      DO 34 K=JBLOCK,NBLOCK
   34 IBLOCK(K)=IBLOCK(K+1)
      IF(LTYP .EQ. 1) RETURN
   36 IF(MODE .NE. 1) GO TO 35
      IF(RHSTYP .EQ. 3 .OR. RHSTYP .EQ. 4) RETURN
      IF(NSTR .LT. 6) RETURN
      JPTR=IPTR-1
      CALL CNVRT
      RETURN
   35 IF(NFUNC .EQ. 0) RETURN
      IF(IFN .EQ. 1) RETURN
      DO 38 J=1,NFUNC
      LOC=FNCLOC(J)
      INDEX=BITGET(IDTBL(3,LOC),32,9)
      KLAS=BITGET(ISUBLT(3,INDEX),10,4)
      IF(KLAS .NE. 1 .AND. KLAS .NE. 2) GO TO 38
      CALL CALL2
      RETURN
   38 CONTINUE
      RETURN
   40 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
   50 CALL ERROR(15,IDM1,IDM2,IDM3,IDM4)
      RETURN
   60 CALL ERROR(89,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
```

```
      SUBROUTINE INTRIN
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      INTEGER BITGET
      DO 100 I=1,NLIST
      IF(BITGET(ISUBLT(3,I),10,4) .NE. 4) GO TO 100
      NXTID(1)=ISUBLT(1,I)
      NXTID(2)=ISUBLT(2,I)
      CALL SEARCH
      CALL COMSCH
      IF(ISRCH(1) .EQ. 1 .OR. ISRCH(3) .EQ. 1)
     $ CALL ERROR(74,NXTID(1),NXTID(2),IDM3,IDM4)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE IO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/INPOUT/NCALL,IN,IOP
      DIMENSION IALPH1(4),IALPH2(5),IALPH3(8)
      INTEGER A,STATRA,RPAR,COMMA,BLANK
      INTEGER BITPUT,BITGET
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /,IC/1HC/
      DATA IALPH1/1HR,1HE,1HA,1HD/
      DATA IALPH2/1HW,1HR,1HI,1HT,1HE/
      DATA IALPH3/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE/
      IFRMT=0
      IF(ITYP .EQ. 12) GO TO 10
      DO 5 I=1,4
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50
    5 CONTINUE
      GO TO 20
   10 DO 15 I=1,5
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50
   15 CONTINUE
   20 IF(NEXT(JPTR) .NE. LPAR) GO TO 50
      CALL GNLE
      IF(JTYP .EQ. 2) GO TO 22
      CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)
      GO TO 28
   22 CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 25
      IDTYP =1
      CALL STORE
      LOC=NID
   25 CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
     $ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=2000+LOC
   28 IF(NEXT(JPTR) .EQ. COMMA) GO TO 40
      JPTR=JPTR-1
      GO TO 30
   40 CALL GNLE
      IF(JTYP .NE. 5) GO TO 26
      CALL STSRCH
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
      GO TO 29
   26 IF(JTYP .NE. 2) GO TO 50
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM1,IDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 27
      IDTYP=1
   CALL STORE
   LOC=NID
27 CALL IMPTYP
   IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1)
  $ CALL ERROR(43,IDM1,IDM2,IDM3,IDM4)
29 IFRMT=1
30 IF(NEXT(JPTR) .NE. RPAR) GO TO 50
   IF(NEXT(JPTR) .NE. BLANK) GO TO 35
   IF(ITYP .EQ. 12 .AND. IFRMT .EQ. 0)
  $ CALL ERROR(44,IDM1,IDM2,IDM3,IDM4)
   GO TO 36
35 JPTR=JPTR-1
   CALL EXPR
   NTYPE=3
   CALL PARSE
   CALL IOSTR
36 IF(MODE .NE. 1 .AND. ITYP .EQ. 11) GO TO 37
   RETURN
37 IF(ITYPE(1) .EQ. 2) GO TO 42
   A(1)=IC
   RETURN
42 WRITE(IOP,45) (A(I),I=1,6),(IALPH3(I),I=1,8)
45 FORMAT(72A1)
   A(1)=IC
   DO 47 I=2,6
47 A(I)=BLANK
   RETURN
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
   RETURN
   END
```

```
      SUBROUTINE IOSTR
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER BITGET
      DO 100 I=1,MARGS
      NBLOCK=NBLOCK+1
      IVR=I
      ICOL=9
      LOC=BITGET(IARGS(IVR),ICOL,9)
      ISUB=BITGET(IARGS(IVR),ICOL+2,2)
      IF(ISUB .EQ. 1) GO TO 90
      IF(ISUB .EQ. 2) GO TO 20
      IF(ITYP .EQ. 11) GO TO 80
      IF(ITYP .EQ. 12) GO TO 90
   20 IDEF=BITGET(IARGS(IVR),ICOL+7,5)
      NMOVE=I-IDEF-1
      DO 30 J=1,NMOVE
      ITEMP=NBLOCK-J
   30 IBLOCK(ITEMP+1)=IBLOCK(ITEMP)
      IBLOCK(ITEMP)=1000+LOC
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=6000+LOC
      GO TO 100
   80 IBLOCK(NBLOCK)=1000+LOC
      GO TO 100
   90 IBLOCK(NBLOCK)=2000+LOC
  100 CONTINUE
      RETURN
      END
```

```
      FUNCTION IPREV(IA)
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER A,BLANK
      DATA BLANK/1H /
      DO 10 I=1,N
      J=IA-I+1
      IF(J .EQ. 0) GO TO 20
      IF(A(J) .EQ. BLANK) GO TO 10
      IPREV=ITYPE(J)
      JPTR=IA-I
      RETURN
   10 CONTINUE
   20 IPREV=3
      JPTR=0
      RETURN
      END
      FUNCTION ITYPE(ID)
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER BITGET
      NXT=NEXT(ID)
      IVAL=BITGET(NXT,8,8)
      IF(IVAL .GE. 193 .AND. IVAL .LE. 201) GO TO 10
      IF(IVAL .GE. 209 .AND. IVAL .LE. 217) GO TO 10
      IF(IVAL .GE. 226 .AND. IVAL .LE. 233) GO TO 10
      IF(IVAL .GE. 240 .AND. IVAL .LE. 249) GO TO 20
      ITYPE=3
      RETURN
   10 ITYPE=1
      RETURN
   20 ITYPE=2
      RETURN
      END
```

```
      SUBROUTINE LOGCHK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LOGIC/LOG,LOGST
      DIMENSION LOGOP(2,11),LOGRA(5),ILOG(2)
      DATA LOGOP/2HLT,1H ,2HLE,1H ,2HGT,1H ,2HGE,1H ,2HEQ,1H ,2HNE,1H ,
     * 2HOR,1H ,3HAND,1H ,3HNOT,1H ,4HTRUE,1H ,4HFALS,1HE/
      DATA IDEC/1H./
      JPTR=LOGST
      DO 10 I=1,6
      NXT=NEXT(JPTR)
      IF(NXT .EQ. IDEC) GO TO 12
   10 LOGRA(I)=NXT
      GO TO 20
   12 IF(I .LT. 3) GO TO 20
      CALL CAA(LOGRA,I-1,ILOG)
      DO 15 I=1,11
      IF(ICOMP(ILOG,LOGOP,I,2) .EQ. 1) GO TO 30
   15 CONTINUE
   20 LOG=0
      RETURN
   30 LOG=1
      LOGID=I
      RETURN
      END
      SUBROUTINE LOGIF
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/STRING/NTYPE,NSTR,STR(500)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,STR,AY,EF
      DATA LPAR/1H(/,AY/1HI/,EF/1HF/
      IF(NEXT(JPTR) .NE. AY) GO TO 110
      IF(NEXT(JPTR) .NE. EF) GO TO 110
      IF(NEXT(JPTR) .NE. LPAR) GO TO 110
      JPTR=JPTR-1
      CALL EXPR
      NSTR=NSTR+1
      STR(NSTR)= -5
      NTYPE=2
      CALL PARSE
      CALL FNCSTR
      CALL BLKSTR
      IF(ITYP .GT. 15) GO TO 130
      LTYP=1
      GO TO (10,20,30,40,50,60,70,80,70,70,90,90,100,100,100),ITYP
   10 CALL INIT
      RETURN
   20 CALL ASSIGN
      RETURN
   30 CALL GOTO
   35 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=998
      NBRNCH=2
      RETURN
   40 CALL ASGOTO
   45 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=998
      NBRNCH=NBRNCH+1
      RETURN
   50 CALL CTGOTO
      GO TO 45
   60 CALL ARIF
      GO TO 45
   70 CALL SIMP
      IF(ITYP .EQ. 7) RETURN
      GO TO 35
   80 CALL CALL
      RETURN
   90 CALL IO
      RETURN
  100 CALL AUXIO
      RETURN
  110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
  130 CALL ERROR(45,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE LOOPCK
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER STATRA,BITGET
      IF(NSTACK .EQ. 0) RETURN
      IBLKST=1
   10 IBLKND=BITGET(IBLOCK(IBLKST),24,12)-1
      LOOP2=BITGET(IBLOCK(IBLKST),12,6)
      NBR=BITGET(IBLOCK(IBLKST),6,6)
      IF(IBLKND .EQ. -1) IBLKND=NBLOCK
      IST=IBLKND-NBR+1
      DO 100 I=IST,IBLKND
      JLOOP=LOOP2
      IF(IBLOCK(I) .GE. 998) GO TO 100
      IBLK=IBLOCK(I)
      NXTBLK=BITGET(STATRA(2,IBLK),32,14)
      KLOOP=BITGET(IBLOCK(NXTBLK),12,6)
      IF(KLOOP .EQ. 0) GO TO 100
      IF(JLOOP .EQ. 0) GO TO 200
   50 IF(JLOOP .EQ. KLOOP) GO TO 100
      JLOOP=ISTACK(3,JLOOP)
      IF(JLOOP .EQ. 0) GO TO 200
      GO TO 50
  200 IBLK=IBLOCK(I)
      WRITE(6,201) STATRA(1,IBLK)
  201 FORMAT(6X,65H ILLEGAL TRANSFER INTO THE RANGE OF A DO LOOP AT STAT
     1EMENT NUMBER,I6)
  100 CONTINUE
      IF(IBLKND .EQ. NBLOCK) RETURN
      IBLKST=IBLKND+1
      GO TO 10
      END
      SUBROUTINE LVDLET
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYP1(10),NSKIP
      INTEGER FLGSPC,FL0MSK,FL1MSK,FL2MSK,FL5MSK,FLG67,REGASP,THIS
     + ,FL3MSK,FL4MSK,SEQSPC
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                      .
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)
      EQUIVALENCE (ILOG,RLOG)
      DATA NFLG02/95/
      KONFLC=0
C     DETERMINE DIRECTION TO PROCEED FOR MULTIVALUE LISTS
      JPOS=IPOS
      IPOS=IABS(IPOS)
      IF(IADD.NE.-1)  GO TO  74
      IF(IARG.EQ.-1) GO TO 66
      IADD=IFUNC+IARG
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE
      IF(AND(FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 99
    1 IF(NODSPC(IADD).EQ.IARG) GO TO 4
C     SEARCH CONFLICT LIST FOR THE FUNCTION
      IADD=LNKSPC(IADD)
      IF(AND(FLGSPC(IADD),FL5MSK) .NE. 0.) GO TO 99
      GO TO 1
   66 IADD=IFUNC
C     TO DELETE A SPECIFIC TYPE OF NODE (INDEXED DELETE), GO TO 72
    4 IF(ITYP.NE.-1) GO TO 72
      IF(AND(FLGSPC(IADD),FL0MSK) .EQ. 0.) GO TO 6
      ISADD=LSTSPC(IADD)
C     DELETE ENTIRE MULTIVALUED FUNCTION, RETRIEVE FIRST VALUE
      IVAL=NODSPC(ISADD)
    5 NXTADD=LSTSPC(ISADD)
      NODSPC(ISADD)=NODSPC(REGASP)
      LSTSPC(ISADD)=REGASP
      LNKSPC(ISADD)=0
      FLGSPC(ISADD)=0
      ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=ISADD
      NODSPC(REGASP)=ISADD
      IF(AND(FLGSPC(NXTADD),FL0MSK) .NE. 0.) GO TO 2
      ISADD=NXTADD
      GO TO 5
C     FUNCTION IS SINGLE VALUED, RETRIEVE VALUE
    6 IVAL=LSTSPC(IADD)
    2 IF(AND(FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 68
      NXFUNC=LNKSPC(IADD)
      IF(AND(FLGSPC(NXFUNC),FL5MSK) .NE. 0.) GO TO 10
      NODSPC(IADD)=NODSPC(NXFUNC)
      LSTSPC(IADD)=LSTSPC(NXFUNC)
      LNKSPC(IADD)=LNKSPC(NXFUNC)
      FLGSPC(IADD)=FLGSPC(NXFUNC)
      RLOG=OR(FLGSPC(IADD),FL5MSK)
      FLGSPC(IADD)=ILOG
      IF(AND(FLGSPC(IADD),FL0MSK) .EQ. 0.) GO TO 9
      KVAL=LSTSPC(IADD)
 8    KVAL=LSTSPC(KVAL)
      ISUB=LSTSPC(KVAL)
      IF(AND(FLGSPC(ISUB),FL0MSK) .EQ. 0.) GO TO 8
      LSTSPC(KVAL)=IADD
 9    IADD=NXFUNC
  10  NODSPC(IADD)=NODSPC(REGASP)

 11   LSTSPC(IADD)=REGASP
      LNKSPC(IADD)=0
      FLGSPC(IADD)=0
      ISUB=LSTSPC(IADD)
      NODSPC(ISUB)=IADD
      ISUB=NODSPC(IADD)
      LSTSPC(ISUB)=IADD
      RETURN
  72  IF(AND(FLGSPC(IADD),FL0MSK) .NE. 0.) GO TO 20
      IF(IPOS.NE.1) GO TO 99
      IF(ITYP.EQ.3) GO TO 6
      RLOG=AND(FLGSPC(IADD),FLG67)
      ISTYP=ILOG
      IF(ISTYP.EQ.ITYP) GO TO 6
 99   IVAL=-1
      RETURN
 20   IND=0
      RLOG= OR(FLGSPC(IADD),FL4MSK)
      FLGSPC(IADD)=ILOG
      LAST=IADD
      IF(JPOS)121,99,21
 121  ISUB=LSTSPC(IADD)
      LAST1=LNKSPC(ISUB)
      THIS=LAST1
      GO TO 27
 21   IF(JPOS.LT.0) GO TO 80
      THIS=LSTSPC(LAST)
      IF(AND(FLGSPC(THIS),FL0MSK) .NE. 0.) GO TO 99
      GO TO 27
 80   THIS=LNKSPC(LAST)
      IF(THIS.EQ.LAST1) GO TO 99
 27   IF(ITYP.EQ.3) GO TO 23
      RLOG=AND(FLGSPC(THIS),FLG67)
      ISTYP=ILOG
      IF(ISTYP.EQ.ITYP)   GO TO 23
 22   LAST=THIS
      GO TO 21
 23   IND=IND+1
      IF(IND.NE.IPOS)     GO TO 22
C    RETRIEVE THE IPOS"TH OF THE KTYP"TH VALUE BEFORE DELETING
      IVAL=NODSPC(THIS)
      MADD=IADD
      IF(JPOS.GT.0) GO TO 55
      NEXT=LNKSPC(THIS)
      IF(THIS.EQ.LAST1) GO TO 82
      LNKSPC(LAST)=NEXT
      GO TO 83
```

```
   82 ISUB=LSTSPC(IADD)
      LNKSPC(ISUB)=NEXT
  83  IF(NEXT.EQ.LAST1) GO TO 84
      LSTSPC(NEXT)=LAST
      GO TO 85
  84  LSTSPC(IADD)=LAST
  85  IADD=THIS
      GO TO 86
  55  NEXT=LSTSPC(THIS)
      IF(AND(FLGSPC(NEXT),FLOMSK) .NE. 0.) GO TO 50
      LNKSPC(NEXT)=LAST
      GO TO 24
   50 ISUB=LSTSPC(IADD)
      LNKSPC(ISUB)=LAST
  24  IADD=THIS
      JLAST=LNKSPC(THIS)
      ISUB=LSTSPC(JLAST)
      IF(AND(FLGSPC(ISUB),FLOMSK) .NE. 0.) LNKSPC(NEXT)=JLAST
      LSTSPC(LAST)=NEXT
  86  KLAST=LSTSPC(MADD)
      IF(LNKSPC(KLAST).NE.KLAST) GO TO 10
C
C     CONVERT TO SINGLE VALUE LIST
C
      LSTSPC(MADD)=NODSPC(KLAST)
      RLOG=AND(OR(FLGSPC(MADD),FLGSPC(KLAST)),NFLG02)
      FLGSPC(MADD)=ILOG
      FLGSPC(KLAST)=0
      LNKSPC(KLAST)=0
      NODSPC(KLAST)=NODSPC(REGASP)
      LSTSPC(KLAST)=REGASP
      ISUB=LSTSPC(KLAST)
      NODSPC(ISUB)=KLAST
      ISUB=NODSPC(KLAST)
      LSTSPC(ISUB)=KLAST
      GO TO 10
   74 IF(AND(FLGSPC(IADD),FLOMSK) .NE. 0.) GO TO 99
      IF(AND(FLGSPC(IADD),FL2MSK) .EQ. 0.) GO TO 99
      LAST=LNKSPC(IADD)
      NEXT=LSTSPC(IADD)
      LSTSPC(LAST)=NEXT
      IF(AND(FLGSPC(NEXT),FLOMSK) .NE. 0.) GO TO 10
      LNKSPC(NEXT)=LAST
      GO TO 10
  68  NEXT=LNKSPC(IADD)
      NEXT1=NEXT
  25  IF(LNKSPC(NEXT1).EQ.IADD)  GO TO 26
      NEXT1=LNKSPC(NEXT1)
      GO TO 25
  26  LNKSPC(NEXT1)=NEXT
      KONFLC=1
      GO TO 10
      END
      SUBROUTINE LVEXIT(N)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSGSP(      1)
      COMMON /TYP/ NN(3),ERRFLG
      COMMON /STRING/ NTYPE,NSTR
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      INTEGER R,RTEMP,STJ,STACK,ASSOC,START,STOP
      COMMON /GIRL/ MM(19),OPRAND
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /NTIMES/ NTIMES,MAXI
      INTEGER STRING,HOL,ACTION,RIGHT,FUNC1,FUNC2,FUNC3,OPRAND
      LOGICAL ERRFLG
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      IF(MAXJ .NE. 0) PRINT 100,MAXJ
  100 FORMAT(1X,44H STATEMENT IS TOO COMPLEX. CORRECT TO CHAR. ,I3)
      IF(MAXJ .EQ. 0) PRINT 200,MAXI,NSTR
  200 FORMAT(1X,29H STATEMENT TOO LONG AT CHAR. ,I3,3H OF,I3)
C COMMENTS USED IN CASE OF GIRS PROBLEMS WHEN MEMORY USED UP
C     GO TO 10
C     IF(MAXJ .EQ. 0) GO TO 50
      IF(MAXJ .EQ. 0) GO TO 10
      DO 30 NCHAR=1,MAXJ
C     STRING+HOL,NCHAR(-LEFT,-RIGHT,-HOL,-STRING)
C****     STRING              +           HOL
      LVVPOS =      NCHAR
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LV1    AAT =      STRING
      IF (LVVAL.NE.-1) LV1    AAT = LVVAL
      LV1    AAX = LV1    AAT
C**** LV1    AAX              -          LEFT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     LEFT
      LVVARG=LV1    AAX
      CALL LVDLET
      LV1    AAX = LV1    AAT
C**** LV1    AAX              -        RIGHT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     RIGHT
      LVVARG=LV1    AAX
      CALL LVDLET
      LV1    AAX = LV1    AAT
C**** LV1    AAX              -           HOL
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=       HOL
      LVVARG=LV1    AAX
      CALL LVDLET
      LV1  AAX = LV1     AAT
C**** LV1  AAX              -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAX
      CALL LVDLET
   30 CONTINUE
C     STRING-STRING
C****     STRING            -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=    STRING
      CALL LVDLET
C     OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)
      LV1  AAT =     OPRAND
C**** LV1  AAT               -      OPRAND
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    OPRAND
      LVVARG=LV1    AAT
      CALL LVDLET
C**** LV1  AAT               -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAT
      CALL LVDLET
C**** LV1  AAT               -      ACTION
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    ACTION
      LVVARG=LV1    AAT
      CALL LVDLET
C**** LV1  AAT               -       FUNC1
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     FUNC1
      LVVARG=LV1    AAT
      CALL LVDLET
   10 CONTINUE
      REWIND 19
      NTIMES=0
C  10 CONTINUE
```

```
C     REWIND 99
C     NTIMES=0
      J=NSTR+1
      R=STOP
      STJ=R
      ERRFLG=.TRUE.
      JSTACK=1
      STACK(JSTACK,1)=STOP
      STACK(JSTACK,2)=100
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C     NSTR=0
      NSTR=MAXJ
C     COMPLETE
      RETURN
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      GO TO 25001
      END
```

```
      SUBROUTINE LVFECH(N)
      INTEGER FLGSPC,SEQSPC,REGASP
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)
      COMMON/LVRAND/KPRIME,KS,KX,KDY,KDX,KTEMP
      READ(N) MEMSZE,REGASP,KPRIME,KS,KX,KTEST,KDY,KTEMP,KDX,KNUM
     +,ISEQSZ
      READ(N)(NODSPC(I),I=1,MEMSZE)
      READ(N)(LSTSPC(I),I=1,MEMSZE)
      READ(N)(LNKSPC(I),I=1,MEMSZE)
      READ(N)(FLGSPC(I),I=1,MEMSZE)
      READ(N)(SEQSPC(I),I=1,ISEQSZ)
      PRINT 10
   10 FORMAT(////1X,34H GRAPH HAS BEEN PLACED INTO MEMORY//)
      RETURN
      END
      SUBROUTINE LVFIND(INDEX,INDXAD,KFUNC,KARG)
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYP1(10),NSKIP
      INTEGER FLGSPC,REGASP,FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,
     + FLG67,SEQSPC
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67
      COMMON /LVTAHL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)
      EQUIVALENCE(RLOG,ILOG)
      DATA NFLAG4/247/
      IADD=IFUNC+IARG
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE
      LSTHED=0
      IF(AND(FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 99
   1  IF(NODSPC(IADD).EQ.IARG)  GO TO 4
      IADD=LNKSPC(IADD)
      IF(AND(FLGSPC(IADD),FL5MSK) .NE. 0.) GO TO 99
      GO TO 1
    4 IF(AND(FLGSPC(IADD),FL0MSK) .NE. 0.) GO TO 14
      RLOG=AND(FLGSPC(IADD),FLG67)
      ISTYP=ILOG
      IF(ITYP.EQ.3) GO TO 11
      IF(ISTYP.EQ.3)ISTYP=2
      IF(ISTYP.NE.ITYP)  GO TO 99
  11  IVAL=LSTSPC(IADD)
      IF((IPOS.NE.1).AND.(IPOS.NE.-1))  GO TO 99
      RLOG=AND(FLGSPC(IADD),FLG67)
      ITYP=ILOG
      LSTHED=-1
      RETURN
  14  LSTHED=IADD
      IND=0
      KNDEX=IABS(INDEX)
      JPOS=IABS(IPOS)
      IF(NSKIP.EQ.1) GO TO 50
      IF((KFUNC.NE.IFUNC).OR.(KARG.NE.IARG)) GO TO 50
      IF(AND(FLGSPC(LSTHED),FL4MSK) .NE. 0.) GO TO 50
      IF((IPOS*INDEX) .LE. 0) GO TO 50
      IF(JPOS.LT.2) GO TO 50
      NDX=FLGSPC(INDXAD)
      IF(AND(NDX,FL5MSK) .NE. 0.) GO TO 50
      IF(AND(NDX,FL1MSK) .EQ. 0.) GO TO 50
      IF(JPOS.GE.KNDEX)  GO TO 25
      IF((JPOS+JPOS).LE.KNDEX)  GO TO 50
      IF(KPOS)30,99,40
   50 RLOG=AND(FLGSPC(LSTHED),NFLAG4)
      FLGSPC(LSTHED)=ILOG
      IF(IPOS) 20,99,10
C
COUNT DOWN FROM THE TOP  OF THE LIST
C
  10  IADD=LSTSPC(IADD)
      IF(AND(FLGSPC(IADD),FL0MSK) .NE. 0.) GO TO 99
      RLOG=AND(FLGSPC(IADD),FLG67)
      ISTYP=ILOG
      IF(ITYP.EQ.3) GO TO 22
      IF(ISTYP.EQ.3)ISTYP=2
      IF(ISTYP.NE.ITYP) GO TO 10
   22 IND=IND+1
      IF(IND.NE.JPOS) GO TO 10
   28 IVAL=NODSPC(IADD)
      RLOG=AND(FLGSPC(IADD),FLG67)
      ITYP=ILOG
   55 INDEX=KPOS
      INDXAD=IADD
      KFUNC=IFUNC
      KARG=IARG
      RETURN
C
COUNT UP FROM THE BOTTOM OF THE LIST
C
   20 IADD=LSTSPC(IADD)
      KTEST=0
   23 IADD=LNKSPC(IADD)
      IF(KTEST.EQ.0) GO TO 24
      ISUB=LSTSPC(IADD)
      IF(AND(FLGSPC(ISUB),FLOMSK) .NE. 0.) GO TO 99
   24 KTEST=1
      RLOG=AND(FLGSPC(IADD),FLG67)
      ISTYP=ILOG
      IF(ITYP.EQ.3) GO TO 21
      IF(ISTYP.EQ.3)ISTYP=2
      IF(ISTYP.NE.ITYP) GO TO 23
   21 IND=IND+1
      IF(IND.NE.JPOS) GO TO 23
   29 IVAL=NODSPC(IADD)
      RLOG=AND(FLGSPC(IADD),FLG67)
      ITYP=ILOG
      GO TO 55
   25 IF(KPOS)40,99,30
C
COUNT DOWN FROM INDXADD
C
   30 JPOS=IABS(JPOS-KNDEX)
      IADD=INDXAD
      IF(JPOS.EQ.0) GO TO 28
      GO TO 10
C
COUNT UP FROM INDXADD
C
   40 JPOS=IABS(JPOS-KNDEX)
      IADD=INDXAD
      IF(JPOS.EQ.0) GO TO 29
      KTEST=1
      GO TO 23
   99 IVAL=-1
      INDEX=0
      INDXAD=0
      KFUNC=0
      KARG=0
      RETURN
      END
      SUBROUTINE LVGRN(NODE)
      INTEGER FLGSPC,REGASP
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)
      COMMON/LVRAND/KPRIME,KSEED,NROW,KDNODE,KDROW,KTEMP
      NODE=KTEMP+KDNODE
      KTEMP=NODE
      KDNODE=KDNODE+1
      IF(NODE .GT. MEMSZE) GO TO 5
      RETURN
    5 IF(NROW .GT. KPRIME) GO TO 10
      NROW=NROW+KSEED
      IF(NROW .GT. KPRIME) NROW=NROW-KPRIME
      NODE=NROW
      KTEMP=NODE
      KDNODE=KPRIME+1
      IF(NODE .NE. KSEED) RETURN
      NROW=0
      KDROW=KPRIME
   10 KDROW=KDROW+1
      NROW=NROW+KDROW
      NODE=NROW
      KTEMP=NODE
      KDNODE=KDROW
      IF(NODE .GT. MEMSZE) GO TO 20
      RETURN
  20  PRINT 15
   15 FORMAT(1H ,1X,47H ERROR...NUMBER OF NODES EXCEEDS REQUEST MEMORY/
     *27H THE PROGRAM IS TERMINATED.)
      STOP
      END
      SUBROUTINE LVNSRT
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP2,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYP(10),NSKIP
      INTEGER FLGSPC,FL0MSK,FL1MSK,FL2MSK,FL5MSK,FLG67,REGASP,TEMP,THIS,
     $ FLGTMP,HEAD,OLDLOC,ASPREG,SEQSPC,FL3MSK,FL4MSK
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)
      EQUIVALENCE (ILOG,RLOG)
      DATA NFLG67/252/
C
      FLGTMP=FL1MSK
C
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98
C
C   FORM FIRST WORD OF SINGLE OR MULTIVALUED FUNCTION
      IF(NVAL.EQ.1)GO TO 20
      LSTTMP=REGASP
      RLOG=OR(FL0MSK, OR(FLGTMP,FL2MSK))
      FLGTMP=ILOG
      GO TO 21
  20  LSTTMP=IVALS(1)
   21 RLOG=OR(FLGTMP,ITYP(1))
      FLGTMP=ILOG
C
C-----------------------------------------------------------------------
C-DETERMINE ADDRESS FOR FUNCTION
      IADD=IFUNC+IARG
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE
C
C    IF THAT ADDRESS IS ALREADY IN WORKING SPACE, GO TO 25
      IF(IDSTRY-1) 125,300,350
  125 IF(AND(FL1MSK,FLGSPC(IADD)) .NE. 0.) GO TO 25
C
C     UPDATE REGASP(IF NECESSARY)
 23   IF(IADD.EQ.REGASP)REGASP=LSTSPC(IADD)
C
C    UPDATE AVAILABLE SPACE
      ISUB=NODSPC(IADD)
      LSTSPC(ISUB)=LSTSPC(IADD)
      ISUB=LSTSPC(IADD)
      NODSPC(ISUB)=NODSPC(IADD)
C
C     INSERT FUNCTION
  24  NODSPC(IADD)=IARG
      LSTSPC(IADD)=LSTTMP
      LNKSPC(IADD)=IADD
      RLOG=OR(FL5MSK,OR(FLGSPC(IADD),FLGTMP))
      RLOG=OR(RLOG,FL4MSK)
      FLGSPC(IADD)=ILOG
C
C   INSERT ANY ADDITIONAL FUNCTION VALUES
      HEAD=IADD
      OLDLOC=IADD
      IF(NVAL.GT.1)GO TO 50
C
C   IF LAST CELL OF AVAILABLE SPACE WAS USED, WRITE MESSAGE
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      IVAL=IABS(IVALS(1))
      RETURN
C
C   IF THAT ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST, GO TO 41
   25 IF(AND(FLGSPC(IADD),FL5MSK) .GT. 0.)  GO TO 41
C
C   IF THAT ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST, GO TO 35
      R1=AND(FL2MSK,FLGSPC(IADD))
      R2=AND(FL0MSK,FLGSPC(IADD))
      IF(R1 .GT. 0. .AND. R2 .EQ. 0.) GO TO 35
C
C-----------------------------------------------------------------------
C-THE ADDRESS CONTAINS A FUNCTION ON A CONFLICT LIST,BUT NOT THE HEAD OF
      THIS=IADD
C
C   FIND THE PRECEDING FUNCTION ON THE CONFLICT LIST
   26 ISUB=LNKSPC(THIS)
      IF(LNKSPC(ISUB) .EQ. IADD) GO TO 27
      THIS=LNKSPC(THIS)
      GO TO 26
 27   LAST=LNKSPC(THIS)
      NEWLOC=REGASP
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98
C
C   UPDATE AVAILABLE SPACE AND REGASP
   28 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)

C   MOVE THE FUNCTION ON A CONFLICT LIST TO THE FIRST CELL OF AVAILABLE
  29  NODSPC(NEWLOC)=NODSPC(IADD)
      LSTSPC(NEWLOC)=LSTSPC(IADD)
      LNKSPC(NEWLOC)=LNKSPC(IADD)
      FLGSPC(NEWLOC)=FLGSPC(IADD)
      FLGSPC(IADD)=0
      LNKSPC(LAST)=NEWLOC
C
C   INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST
      NODSPC(IADD)=IARG
      LNKSPC(IADD)=IADD
      LSTSPC(IADD)=LSTTMP
      RLOG=OR(FL5MSK,OR(FLGSPC(IADD),FLGTMP))
      RLOG=OR(RLOG,FL4MSK)
      FLGSPC(IADD)=ILOG
      IF(AND(FLGSPC(NEWLOC),FL0MSK) .EQ. 0.) GO TO 34
C
C   IF THE FUNCTION THAT WAS MOVED IS THE HEAD OF A MULTIVALUE LIST,FIX
      NEXT=LSTSPC(NEWLOC)
 30   NEXT=LSTSPC(NEXT)
```

```
      IF(LSTSPC(NEXT).NE.IADD)GO TO 30
      LSTSPC(NEXT)=NEWLOC
C
C    INSERT ANY ADDITIONAL FUNCTION VALUES
   34 HEAD=IADD
      OLDLOC=IADD
      IF(NVAL.GT.1)GO TO 50
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      IVAL=IABS(IVALS(1))
      RETURN
C
C-----------------------------------------------------------------------
C-THE ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST
   35 NEWLOC=REGASP
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98
C
C    UPDATE AVAILABLE SPACE AND REGASP
    36 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
C
C    MOVE THE VALUE ON A MULTIVALUE LIST TO THE FIRST CELL OF AVAILABLE S
   37 NODSPC(NEWLOC)=NODSPC(IADD)
      LSTSPC(NEWLOC)=LSTSPC(IADD)
      LNKSPC(NEWLOC)=LNKSPC(IADD)
      FLGSPC(NEWLOC)=FLGSPC(IADD)
      FLGSPC(IADD)=0
C
C     RESET POINTERS
C
      L1=LSTSPC(NEWLOC)
      IF(AND(FLOMSK,FLGSPC(L1)) .EQ. 0.) GO TO 200
      ISUB=LSTSPC(L1)
      LNKSPC(ISUB)=NEWLOC
      GO TO 201
  200 LNKSPC(L1)=NEWLOC
   201 ISUB=LNKSPC(NEWLOC)
      KZVAL=LSTSPC(ISUB)
      IF(AND(FLGSPC(KZVAL),FLOMSK) .NE. 0.)  GO TO 38
      ISUB=LNKSPC(NEWLOC)
      LSTSPC(ISUB)=NEWLOC
      GO TO 39
   38 LSTSPC(KZVAL)=NEWLOC
   39 NODSPC(IADD)=IARG
C    INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST
      LNKSPC(IADD)=IADD
      LSTSPC(IADD)=LSTTMP
      RLOG=OR(FL5MSK,OR(FLGSPC(IADD),FLGTMP))
      RLOG=OR(RLOG,FL4MSK)
      FLGSPC(IADD)=ILOG
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      IVAL=IABS(IVALS(1))
      RETURN
```

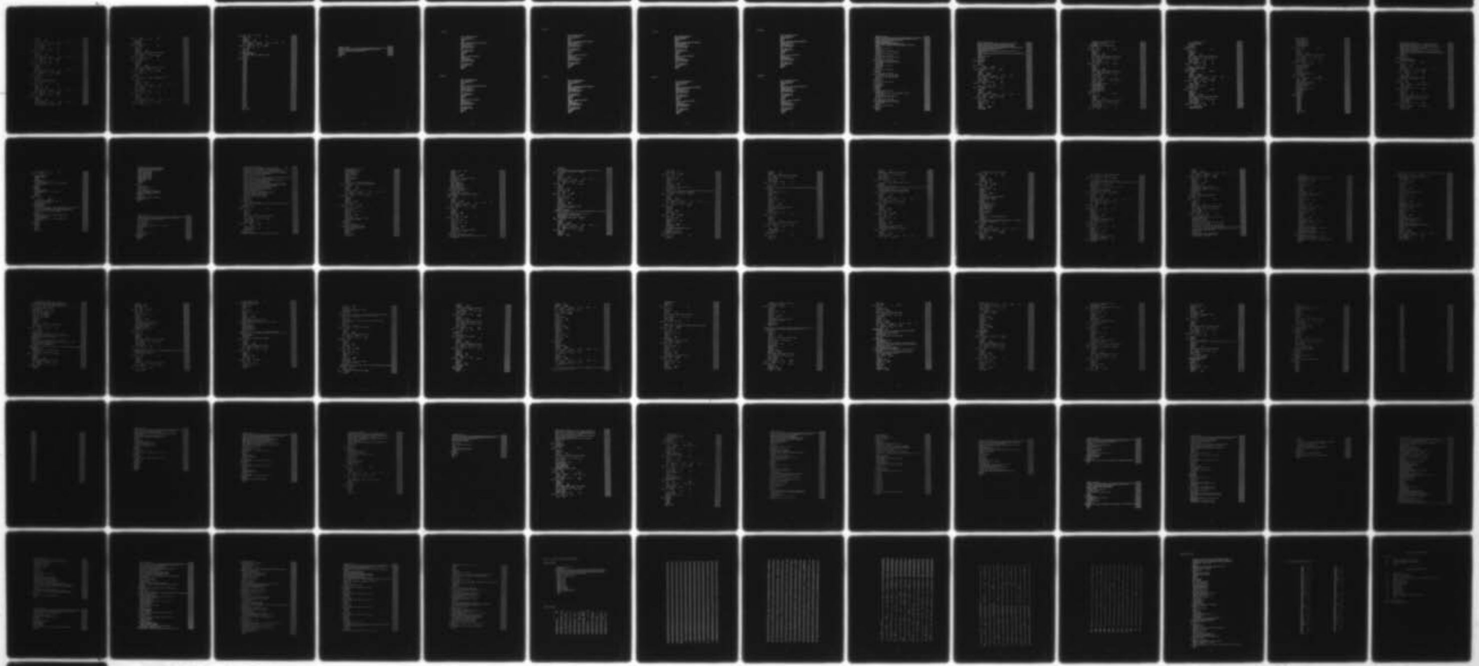
```
C
C-----------------------------------------------------------------------
C-THE ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST
 41   THIS=IADD
C
C    IF THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST, GO TO 60
  42  IF(NODSPC(THIS).EQ.IARG)GO TO 43
      ISUB=LNKSPC(THIS)
      IF(AND(FLGSPC(ISUB),FL5MSK) .NE. 0.) GO TO 60
      THIS=LNKSPC(THIS)
      GO TO 42
C
C-----------------------------------------------------------------------
C-THE FUNCTION TO BE INSERTED IS ON THE CONFLICT LIST
  43  HEAD=THIS
      IF(AND(FLGSPC(THIS),FL0MSK) .EQ. 0.) GO TO 51
      NEXT=LSTSPC(THIS)
C
C     OLDLOC IS THE LOCATION OF THE LAST VALUE ON THE MULTIVALUE LIST
C
      OLDLOC=LNKSPC(NEXT)
C
C-----------------------------------------------------------------------
C-INSERT ADDITIONAL FUNCTION VALUES
 50   LSTASP=NODSPC(REGASP)
      IN=0
      GO TO 56
C
C-----------------------------------------------------------------------
C-FORM MULTIVALUE LIST TO ADD VALUE(S) TO SINGLE-VALUED FUNCTION
  51  IN=0
      IF(REGASP.EQ.LSTSPC(REGASP))GO TO 98
      LSTASP=NODSPC(REGASP)
      NEWLOC=REGASP
      REGASP=LSTSPC(REGASP)
      NODSPC(NEWLOC)=LSTSPC(THIS)
      RLOG=OR(FLGSPC(NEWLOC),AND(FLGSPC(THIS),FLG67))
      FLGSPC(NEWLOC)=ILOG
      RLOG=OR(FL0MSK,OR(FL2MSK,AND(NFLG67,FLGSPC(THIS))))
      FLGSPC(THIS)=ILOG
      OLDLOC=THIS
C
C-----------------------------------------------------------------------
C    INSERT ANOTHER VALUE ON MULTIVALUE LIST
   52 RLOG=OR(FL1MSK,OR(FL2MSK,FLGSPC(NEWLOC)))
      FLGSPC(NEWLOC)=ILOG
      LSTSPC(OLDLOC)=NEWLOC
      LNKSPC(NEWLOC)=OLDLOC
 55   OLDLOC=NEWLOC
 56   NEWLOC=REGASP
      IF(IN.GT.0)GO TO 57
C
C    NO VALUES HAVE BEEN INSERTED YET
      IN=1
      GO TO 58
```

```
C
C   SOME VALUES HAVE BEEN INSERTED
 57   IF(IN.EQ.NVAL)GO TO 67
      IN=IN+1
C
   58 IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
  581 REGASP=LSTSPC(REGASP)
  582 NODSPC(NEWLOC)=IVALS(IN)
      RLOG=OR(ITYP(IN),FLGSPC(NEWLOC))
      FLGSPC(NEWLOC)=ILOG
      GO TO 52
C
C   END MULTIVALUE LIST AND UPDATE AVAILABLE SPACE
 67   LSTSPC(OLDLOC)=HEAD
      NODSPC(REGASP)=LSTASP
      LSTSPC(LSTASP)=REGASP
      IVAL=IABS(IVALS(1))
      ISUB=LSTSPC(HEAD)
      LNKSPC(ISUB)=OLDLOC
      NVAL=IN
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      RETURN
C
C-----------------------------------------------------------------------
C-THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST
 60   ASPREG=REGASP
      LSTASP=NODSPC(REGASP)
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98
C
C   UPDATE AVAILABLE SPACE AND REGASP
  601 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
C
C   INSERT FUNCTION IN FIRST CELL OF AVAILABLE SPACE
 61   NODSPC(ASPREG)=IARG
      IF(NVAL.EQ.1)GO TO 611
      LSTSPC(ASPREG)=REGASP
      RLOG=OR(FL0MSK,OR(FL2MSK,FLGSPC(ASPREG)))
      FLGSPC(ASPREG)=ILOG
      GO TO 612
 611  LSTSPC(ASPREG)=IVALS(1)
  612 RLOG=OR(FL1MSK,OR(ITYP(1),FLGSPC(ASPREG)))
      RLOG=OR(RLOG.FL4MSK)
      FLGSPC(ASPREG)=ILOG
      LNKSPC(ASPREG)=LNKSPC(THIS)
      LNKSPC(THIS)=ASPREG
      IF(NVAL.EQ.1)GO TO 613
C
C   INSERT ADDITIONAL VALUES
      LSTASP=NODSPC(REGASP)
      OLDLOC=ASPREG
      HEAD=ASPREG
      IN=0
      GO TO 56
 613  IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      IVAL=IABS(IVALS(1))
      RETURN
C
C     DESTRUCTIVE INSERTION
C
 350  IADD1=IADD
      INDEX=0
      CALL LVFIND(INDEX,INDEX,INDEX,INDEX)
      RLOG=OR(FLGSPC(IADD),FL4MSK)
      FLGSPC(IADD)=ILOG
      IF(IVAL.EQ.-1) GO TO 90
      IF(LSTHED) 354,90,356
 354  LSTSPC(IADD)=IVALS(1)
      GO TO 365
  356 NODSPC(IADD)=IVALS(1)
  365 RLOG=OR(ITYP(1),AND(NFLG67,FLGSPC(IADD)))
      FLGSPC(IADD)=ILOG
      GO TO 360
  90  IF(IPOS)91,99,92
  91  IPOS=IPOS+1
      GO TO 93
  92  IPOS=IPOS-1
  93  IADD=IADD1
      IF(IPOS.EQ.0) GO TO 125
      INDEX=0
      CALL LVFIND(INDEX,INDEX,INDEX,INDEX)
      IF(IVAL.EQ.-1) GO TO 99
      IF(IPOS .LT. 0) GO TO 370
      IADD=IADD1
      GO TO 125
 370  NEWLOC=REGASP
      IF(LSTHED) 325,99,375
C     UPDATE AVAILABLE SPACE
  375 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
      GO TO 377
C
C     NONDESTRUCTIVE INSERTION
C
  300 IADD1=IADD
      NEWLOC=REGASP
      INDEX=0
      CALL LVFIND(INDEX,INDEX,INDEX,INDEX)
      RLOG=OR(FLGSPC(IADD),FL4MSK)
      FLGSPC(IADD)=ILOG
      IF(IVAL.EQ.-1) GO TO 90
      IF(LSTHED) 344,90,346
  344 IF(IPOS .GT. 0) GO TO 325
      IADD=IADD1
```

AD-A043 925 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
MAINTENANCE MANUAL FOR AUDIT. A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)
AUG 77 R J WYBRANIEC, R REGEN

UNCLASSIFIED DTNSRDC-77-0075-VOL-4 NL

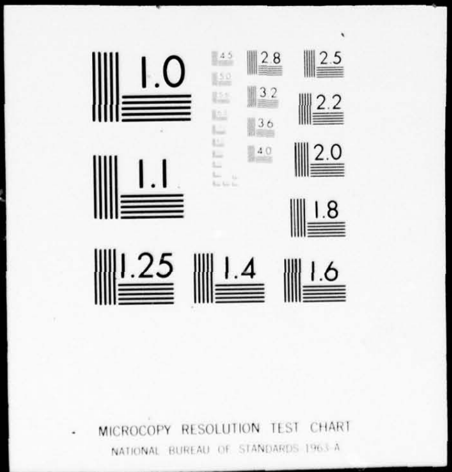
2 OF 2
ADA043925



END
DATE
FILMED
10-77
DDC

2 OF 2

ADA043925



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
      GO TO 125
C     CREATE MULTIVALUE LIST
  325 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
      NWLOC2=REGASP
C     UPDATE AVAILABLE SPACE
      ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
      NODSPC(NEWLOC)=IVALS(1)
      LSTSPC(NEWLOC)=NWLOC2
      LNKSPC(NEWLOC)=NWLOC2
      RLOG=OR(FLGTMP,FL2MSK)
      FLGSPC(NEWLOC)=ILOG
      NODSPC(NWLOC2)=LSTSPC(IADD)
      LSTSPC(NWLOC2)=IADD
      LNKSPC(NWLOC2)=NEWLOC
      R1=AND(FLGSPC(IADD),FLG67)
      RLOG=OR(R1,OR(FL1MSK,FL2MSK))
      FLGSPC(NWLOC2)=ILOG
      LSTSPC(IADD)=NEWLOC
      RLOG=OR(FL2MSK,OR(FL0MSK,FLGSPC(IADD)))
      FLGSPC(IADD)=ILOG
  320 IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909
  360 IVAL=IABS(IVALS(1))
      RETURN
C     UPDATE AVAILABLE SPACE
  346 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)
      IF(IPOS .LT. 0) GO TO 347
 377  ISTLOC=LNKSPC(IADD)
      NODSPC(NEWLOC)=IVALS(1)
      LSTSPC(NEWLOC)=IADD
      LNKSPC(NEWLOC)=ISTLOC
      RLOG=OR(FLGTMP,FL2MSK)
      FLGSPC(NEWLOC)=ILOG
      ISUB=LSTSPC(ISTLOC)
      IF(AND(FLGSPC(ISUB),FL0MSK) .EQ. 0.) GO TO 321
      ISUB=LSTSPC(ISTLOC)
      LSTSPC(ISUB)=NEWLOC
      GO TO 322
  321 LSTSPC(ISTLOC)=NEWLOC
  322 LNKSPC(IADD)=NEWLOC
      GO TO 320
  347 NODSPC(NEWLOC)=IVALS(1)
      LSTSPC(NEWLOC)=LSTSPC(IADD)
      LNKSPC(NEWLOC)=IADD
      RLOG=OR(FLGTMP,FL2MSK)
      FLGSPC(NEWLOC)=ILOG
      ISUB=LSTSPC(IADD)
      IF(AND(FLGSPC(ISUB),FL0MSK) .EQ. 0.) GO TO 323
      KZVAL=LSTSPC(IADD)
      ISUB=LSTSPC(KZVAL)
      LNKSPC(ISUB)=NEWLOC
      GO TO 324
  323 ISUB=LSTSPC(IADD)
      LNKSPC(ISUB)=NEWLOC
  324 LSTSPC(IADD)=NEWLOC
      GO TO 320
  98  IVAL=-3
      PRINT 20001
20001 FORMAT(1X,78H ERROR...THERE IS NO ADDITIONAL SPACE FOR THE GRAPH,
     *THE PROGRAM IS TERMINATED)
      STOP
  99  IVAL=-1
 2    FORMAT(6H ONLY ,I4,28H VALUE(S) HAVE BEEN INSERTED)
 22   FORMAT(1X,I5,1H(,I5,35H) USED LAST CELL OF AVAILABLE SPACE)
      RETURN
 909  PRINT 22,IFUNC,IARG
C     THIS INSERTION HAS FILLED GIRS MEMORY - CALL A USER SUPPLIED
C     PROGRAM - LVEXIT.
      IVAL=-1
      RETURN
      END
      SUBROUTINE LVSETP
      INTEGER FLGSPC,FLAGSP,REGASP,BINFIL,FL0MSK,FL1MSK,FL2MSK,FL5MSK,
     + FL3MSK,FL4MSK,FLG67,SEQSPC
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67
      COMMON/LVVTR5/BINFIL,KOMPAN,NODESP(1)/LVVTR6/LISTSP(1)
     + /LVVTR7/LINKSP(1)/LVVTR8/FLAGSP(1)
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)
      COMMON/LVRAND/KPRIME,KSEED,NROW,KDNODE,KDROW,KTEMP
      FL0MSK=128
      FL1MSK=64
      FL2MSK=32
      FL5MSK=4
      FLG67=3
      FL3MSK=16
      FL4MSK=8
      KSEED=KPRIME/2
      NROW=KSEED
      KTEMP=KSEED-KPRIME
      KDNODE=KPRIME
      REGASP=1
      DO 10 I=2,MEMSZE
      LNKSPC(I)=0
      FLGSPC(I)=0
      NODSPC(I)=I-1
   10 LSTSPC(I-1)=I
      FLGSPC(1)=0
      LNKSPC(1)=0
      NODSPC(1)=MEMSZE
      LSTSPC(MEMSZE)=1
      RETURN
      END
      SUBROUTINE MODID(MODE)
      DIMENSION IBUF(80),IEND(3)
      IF(MODE .NE. 0) GO TO 5
      WRITE(13,1)
    1 FORMAT(5X,16H OUTPUT DEVICE X)
      WRITE(14,2)
    2 FORMAT(5X,16H OUTPUT DEVICE Y)
      WRITE(15,3)
    3 FORMAT(5X,16H OUTPUT DEVICE Z)
    5 DO 10 I=10,12
      ENDFILE I
      REWIND I
      READ(I,7,END=8) ICHAR
    7 FORMAT(A1)
      IEND(I-9)=0
      GO TO 10
    8 IEND(I-9)=1
   10 CONTINUE
      IEOF=0
      DO 15 I=1,3
      IF(IEND(I) .NE. 0) GO TO 15
      IF(IEOF .EQ. 1) GO TO 40
      IEOF=1
      IOUT=9+I
   15 CONTINUE
      IF(IEOF .EQ. 0) GO TO 50
      REWIND IOUT
      IOUT2=IOUT+3
      WRITE(IOUT2,20) MODE
   20 FORMAT(//20X,12H MODE INDEX=,I3)
      DO 30 I=1,100
      READ(IOUT,25,END=60) (IBUF(J),J=1,80)
   25 FORMAT(80A1)
   30 WRITE(IOUT2,25) (IBUF(J),J=1,80)
   40 WRITE(6,45)
   45 FORMAT(5X,73H***** ERROR IN ROLL CALL CHECK - MORE THAN 1 OUTPUT D
     *EVICE WAS WRITTEN ON)
      GO TO 60
   50 WRITE(6,55)
   55 FORMAT(5X,66H***** ERROR IN ROLL CALL CHECK - NO OUTPUT DEVICES WE
     *RE WRITTEN ON)
   60 REWIND 10
      REWIND 11
      REWIND 12
      RETURN
      END
      FUNCTION NEXT(IA)
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER A,BLANK
      DATA BLANK/1H /
      IF(IA .GT. N) GO TO 15
      DO 10 I=IA,N
      IF(A(I) .EQ. BLANK) GO TO 10
      NEXT=A(I)
      IA=I+1
      RETURN
   10 CONTINUE
      NEXT=BLANK
      IA=N+1
      RETURN
   15 NEXT=BLANK
      IA=IA+1
      RETURN
      END
```

```
      FUNCTION NXTBLK(ILOC,IEND)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/LABELS/STATRA(2,200),NLABEL
      INTEGER STATRA,BITGET
      I=IBLOCK(ILOC)
      IF(I .EQ. 999) GO TO 10
      IF(I .EQ. 998) GO TO 5
      NXTBLK=BITGET(STATRA(2,I),32,14)
      RETURN
    5 NXTBLK=IEND+1
      IF(NXTBLK .GT. NBLOCK) CALL ERROR(38,IDM1,IDM2,IDM3,IDM4)
      RETURN
   10 NXTBLK=0
      RETURN
      END
      SUBROUTINE PARSE
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4),
      COMMON/FUNC/  NARY(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG
      COMMON /STRING/ NTYPE,NSTR,STR
      COMMON /GIRL/NTERMS,PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,
     +LE,GT,GE,EQ,NE,OR,AND,NOT,EQUALS,OPRAND
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON /NTIMES/ NTIMES,I
      COMMON/VAR/VFOR(30),NUMCHR,NCHRP,CHAR,NDICT
      INTEGER TYPE1,TYPE2,START,TYP(3)
      LOGICAL ERRFLG,FAIL
      INTEGER STR(500),STEMP,ST,DICT(19)
      EQUIVALENCE(DICT(1),PLUS)
      INTEGER PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,LE,GT,GE,EQ,
     +NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT
     +,STRING,FUNC1,FUNC2,FUNC3
      IF(NTIMES .GT. 0) GO TO 3
      NTIMES=1
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      CALL PHONEY
      CALL LVFECH(19)
      READ(19)PLUS,MINUS,SLASH,LPAR,PPAR,COMMA,STAR,EXP,LT,LE,GT,GE,EQ,
     +NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT
     +,STRING,FUNC1,FUNC2,FUNC3,NTERMS,(TYP(I),I=1,3)
    3 IF(NSTR .LE. 0) RETURN
      ERRFLG=.FALSE.
      START=TYP(NTYPE)
      NARGS=0
      MAXJ=0
      NOPAR=0
      TYPE1=-1
      TYPE2=-1
      NARRAY=-1
      NDEPTH=0
      NDEP=0
      NFLAG=0
      DO 20 I=1,50
   20 IARGS(I)=0
      DO 22 J=1,22
      DO 22 I=1,5
   22 NARY(I,J)=0
      DO 10 I=1,NSTR
C     STRING(+HOL,I "NTEMP//4,HOL $ "NTEMP)
      LV1     AAB =      STRING
C**** LV1     AAB              +            HOL
      LVVPOS =           I
      LVVTYP =    3
      LVFUNC=      HOL
      LVVARG= LV1   AAB
      CALL LVFIND(LV2    A,LV2     B,LV2     C,LV2     D)
      LV1   AAC = LV1   AAB
      IF (LVVAL.NE.-1) LV1   AAC = LVVAL
C**** LV1   AAC         "       NTEMP
           NTEMP = LV1   AAC
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO         4
      CALL LVGRN(LV1   AAC)
C**** LV1   AAB        HOL  LV1   AAC
      LVDEST=  0
      LVTYPE(1) =  0
      LVVALS(1) = LV1   AAC
      LVVNVL =  1
      LVFUNC =      HOL
      LVVARG=LV1   AAB
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C**** LV1   AAC         "       NTEMP
           NTEMP = LV1   AAC
    4 CONTINUE
      IF(ERRFLG) GO TO 25
      ST=IABS(STR(I))
      IF(STR(I) .LT. 0) GO TO 6
C     NTEMP HOL """ST"""
C****      NTEMP        HOL         """
      LVDEST=  0
      LV1   AAB = ST
      LVTYPE(1) =  1
      LVVALS(1) = LV1   AAB
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      HOL
      LVVARG=    NTEMP
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(ERRFLG) GO TO 25
C     STRING STRING OPRAND//10
C****     STRING      STRING      OPRAND
      LVDEST=  0
      LVTYPE(1) =  0
      LVVALS(1) =     OPRAND
      LVVNVL =  1
      LVFUNC =    STRING
      LVVARG=    STRING
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO        10
      IF(ERRFLG) GO TO 25
    6 STEMP=DICT(ST)
C     STRING STRING ""STEMP""
C****     STRING       STRING           ""
      LVDEST=  0
      LV1     AAC = STEMP
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAC
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=     STRING
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(ERRFLG) GO TO 25
   10 CONTINUE
      LV1     AAD =       OPRAND
      LVDEST=  0
      LV1     AAE = 0
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAE
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC2
      LVVARG=LV1     AAD
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1     AAF = 0
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAF
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC3
      LVVARG=LV1     AAD
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1     AAG = 0
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAG
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       LEVEL
      LVVARG=LV1     AAD
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      CALL RECOG(FAIL)
      IF(FAIL) GO TO 40
   25 CONTINUE
      IF(ERRFLG) PRINT 100
```

```
  100 FORMAT(/)
      CALL PRNTS
      NCHAR=0
   30 NCHAR=NCHAR+1
C     STRING+HOL.NCHAR/35(-LEFT.-RIGHT.-HOL.-STRING/30/30)
C****      STRING           +         HOL
      LVVPOS =       NCHAR
      LVVTYP =    3
      LVFUNC=         HOL
      LVVARG=      STRING
      CALL LVFIND(LV2       E.LV2       F.LV2       G.LV2       H)
      LV1    AAD =      STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          35
      LV1    AAE = LV1    AAD
C**** LV1    AAE           -         LEFT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=       LEFT
      LVVARG=LV1    AAE
      CALL LVDLET
      LV1    AAE = LV1    AAD
C**** LV1    AAE           -        RIGHT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=      RIGHT
      LVVARG=LV1    AAE
      CALL LVDLET
      LV1    AAE = LV1    AAD
C**** LV1    AAE           -         HOL
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=        HOL
      LVVARG=LV1    AAE
      CALL LVDLET
      LV1    AAE = LV1    AAD
C**** LV1    AAE           -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=LV1    ' AAE
      CALL LVDLET
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          30
      IF (LVVTR.NE.-1) GO TO          30
   35 CONTINUE
C     STRING-STRING
C****      STRING           -      STRING
```

```
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=    STRING
      CALL LVDLET
C     OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)
      LV1    AAD =     OPRAND
C**** LV1    AAD            -      OPRAND
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    OPRAND
      LVVARG=LV1     AAD
      CALL LVDLET
C**** LV1    AAD            -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1     AAD
      CALL LVDLET
C**** LV1    AAD            -      ACTION
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    ACTION
      LVVARG=LV1     AAD
      CALL LVDLET
C**** LV1    AAD            -      FUNC1
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    FUNC1
      LVVARG=LV1     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    FUNC2
      LVVARG=LV1     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    FUNC3
      LVVARG=LV1     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    LEVEL
      LVVARG=LV1     AAD
      CALL LVDLET
      NSTR=NCHRP
      RETURN
   40 PRINT 300.MAXJ
  300 FORMAT(1X.34H PARSE FAILED AFTER CHARACTER NO. .I3)
      ERRFLG=.TRUE.
      GO TO 25
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      GO TO 25001
      END
```

```
      SUBROUTINE PHONEY
      INTEGER FLGSPC.FLAGSP
      COMMON/LVVTR1/LVVSZE.LVVGSP.NODSPC(1000)/LVVTR2/LSTSPC(1000)
      COMMON/LVVTR3/LNKSPC(1000)/LVVTR4/FLGSPC(1000)
      COMMON/LVVTR5/LVFILE.LVCMPR.NODESP(      1)
     */LVVTR6/LISTSP(      1)/LVVTR7/LINKSP(      1)
     */LVVTR8/FLAGSP(      1)
      COMMON/LVRAND/LVKPRM.LVKS.LVKX.LVKDY.LVKDX.LVTEMP
      COMMON/LVARGS/LVFUNC.LVVARG.LVVAD.LVVPOS.LVVTYP.LVVAL.
     *LVHEAD.LVVNVL.LVDEST.LVVALS(10).LVTYPE(10).LVSKIP
      COMMON/LVTABL/LVTSIZ.LVMAP(      1)/LVVSEQ/LVSIZE.LVSQSP(      1)
C     EXECUTE
      LVVSZE=1000
      LVFILE=0
      LVCMPR= 0
      LVSIZE=      1
      LVSKIP=1
      LVKPRM=17
      CALL LVSETP
      RETURN
      END
      SUBROUTINE PPNTS
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      COMMON/VAR/VFOR,NCHAR,NCHARP,CHAR,NDICT
      COMMON/TYP/NARRAY,TYPE1,TYPE2,ERRFLG,LHSTYP
      COMMON /STRING/ NTYPE,NSTR,STR
      COMMON /GIPL/ NAN(19),OPRAND
      LOGICAL ERRFLG
      INTEGER VFOR(30),CHAR,STRING,HOL,RIGHT,STR(1),OPRAND
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      NCHAR=0
      NCHARP=0
      DO 5 I=1,30
    5 VFOR(I)=0
      NINT=1
      NTMP=1
      DO 10 I=1,NSTR
C     STRING(+HOL.I "NODE,+STRING.I "N1=OPRAND/16)
      LV1  AAD =      STRING
C**** LV1  AAD           +         HOL
      LVVPOS =          I
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG= LV1    AAD
      CALL LVFIND(LV2       A,LV2        B,LV2        C,LV2        D)
      LV1  AAE = LV1      AAD
      IF (LVVAL.NE.-1) LV1      AAE = LVVAL
C**** LV1  AAE           "         NODE
            NODE = LV1      AAE
C**** LV1  AAD           +      STRING
      LVVPOS =          I
      LVVTYP=3
      LVFUNC=     STRING
      LVVARG= LV1    AAD
      CALL LVFIND(LV2       E,LV2        F,LV2        G,LV2        H)
      LV1  AAE = LV1      AAD
      IF (LVVAL.NE.-1) LV1      AAE = LVVAL
C**** LV1  AAE           "         N1
              N1 = LV1      AAE
C****         N1           =      OPRAND
      LVVAL = -100
      IF (        N1.NE.    OPRAND) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         16
      NINT=NINT+1
   16 J=0
   20 J=J+1
C     NODE+LEFT.J/30 (+HOL.1 "CHAR,+HOL.2 "NDICT)
C****         NODE         +         LEFT
      LVVPOS =          J
      LVVTYP=3
      LVFUNC=        LEFT
      LVVARG=        NODE
      CALL LVFIND(LV2      I.LV2      J.LV2       K.LV2      L)
      LV1    AAD =        NODE
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO        30
      LV1    AAE = LV1    AAD
C**** LV1    AAE          +         HOL
      LVVPOS =          1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LV1     AAE
      CALL LVFIND(LV2      M.LV2      N.LV2       O.LV2      P)
      LV1    AAF = LV1    AAE
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL
C**** LV1    AAF          "        CHAR
            CHAR = LV1    AAF
      LV1    AAF = LV1    AAD
C**** LV1    AAF          +         HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LV1     AAF
      CALL LVFIND(LV2      Q.LV2      R.LV2       S.LV2      T)
      LV1    AAE = LV1    AAF
      IF (LVVAL.NE.-1) LV1    AAE = LVVAL
C**** LV1    AAE          "       NDICT
           NDICT = LV1    AAE
      CALL FORM
      GO TO 20
   30 CONTINUE
      IF(NINT .GT. NTMP) GO TO 35
C     N1(+HOL.1 "CHAR,+HOL.2 "NDICT)
      LV1    AAD =         N1
C**** LV1    AAD          +         HOL
      LVVPOS =          1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LV1     AAD
      CALL LVFIND(LV2      U.LV2      V.LV2       W.LV2      X)
      LV1    AAE = LV1    AAD
      IF (LVVAL.NE.-1) LV1    AAE = LVVAL
C**** LV1    AAE          "        CHAR
            CHAR = LV1    AAE
C**** LV1    AAD          +         HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LV1     AAD
      CALL LVFIND(LV2      Y.LV2      Z.LV2       0.LV2      1)
      LV1    AAE = LV1    AAD
      IF (LVVAL.NE.-1) LV1    AAE = LVVAL
```

```
C**** LV1    AAE                "         NDICT
            NDICT = LV1    AAE
      CALL FORM
      GO TO 37
   35 NTMP=NINT
C     NODE+HOL "NDICT
C****        NODE               +            HOL
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=          HOL
      LVVARG=         NODE
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =         NODE
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
C**** LV1    AAD                "         NDICT
            NDICT = LV1    AAD
C     OPRAND+HOL "CHAR
C****      OPRAND               +            HOL
      LVVTYP = 3
      LVVPOS = 1
      LVINDX=0
      LVFUNC=          HOL
      LVVARG=      OPRAND
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =      OPRAND
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
C**** LV1    AAD                "          CHAR
             CHAR = LV1    AAD
      CALL FORM
   37 J=0
   40 J=J+1
C     NODE+RIGHT.J/10 (+HOL.1 "CHAR,+HOL.2 "NDICT)
C****         NODE              +          RIGHT
      LVVPOS =            J
      LVVTYP =    3
      LVFUNC=        RIGHT
      LVVARG=         NODE
      CALL LVFIND(LV2       2,LV2       3,LV2       4,LV2       5)
      LV1    AAD =         NODE
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          10
      LV1    AAE = LV1    AAD
C**** LV1    AAE                +           HOL
      LVVPOS =            1
      LVVTYP=3
      LVFUNC=          HOL
      LVVARG= LV1    AAE
      CALL LVFIND(LV2       6,LV2       7,LV2       8,LV2       9)
      LV1    AAF = LV1    AAE
      IF (LVVAL.NE.-1) LV1     AAF = LVVAL
C**** LV1    AAF                "          CHAR
             CHAR = LV1    AAF
      LV1    AAF = LV1      AAD
C**** LV1    AAF            +          HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG= LV1    AAF
      CALL LVFIND(LV2      AA,LV2      AB,LV2      AC,LV2      AD)
      LV1    AAE = LV1      AAF
      IF (LVVAL.NE.-1) LV1      AAE = LVVAL
C**** LV1    AAE            "          NDICT
            NDICT = LV1     AAE
      CALL FORM
      GO TO 40
   10 CONTINUE
      NC=1+(NCHARP-1)/4
  100 FORMAT(1X,30A4)
      IF(ERRFLG) PRINT 100,(VFOR(I),I=1,NC)
C     COMPLETE
      RETURN
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      LV2I=0
      LV2J=0
      LV2K=0
      LV2L=0
      LV2M=0
      LV2N=0
      LV2O=0
      LV2P=0
      LV2Q=0
      LV2R=0
      LV2S=0
      LV2T=0
      LV2U=0
      LV2V=0
      LV2W=0
      LV2X=0
      LV2Y=0
      LV2Z=0
      LV20=0
      LV21=0
      LV22=0
      LV23=0
      LV24=0
      LV25=0
      LV26=0
      LV27=0
      LV28=0
      LV29=0
      LV2AA=0
      LV2AB=0
      LV2AC=0
      LV2AD=0
      GO TO 25001
      END
```

```
 SUBROUTINE PROG
 COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
 IDTYP=2
 CALL STORE
 RETURN
 END
```

31 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFFE/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFE/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFE/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

30 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFFC/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFC/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFC/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

29 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF8/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF8/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF8/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

28 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF0/
Q1REAL=AND(X,R)
RETURN
EN)
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

27 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFE0/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFE0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFE0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

26 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFC0/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFC0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFC0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

25 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFF80/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFF80/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFF80/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

24 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFF00/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFF00/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFF00/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

```
      SUBROUTINE REALCK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LOGIC/LOG,LOGST
      COMMON/REALNO/IREAL,IRELND,IP
      INTEGER A,DECPT,EEE,PLUS,MINUS,DEE,ELOC
      DATA DECPT/1H./,EEE/1HE/,PLUS/1H+/,MINUS/1H-/,DEE/1HD/
      IDES=0
      IRELND=0
      IF(IP .GE. N) GO TO 90
      JPTR=IP
      IF(NEXT(JPTR) .EQ. DECPT) GO TO 5
      JPTR=IP
      IF(ITYPE(JPTR) .EQ. 2) GO TO 10
      GO TO 90
    5 IF(JPTR .GT. N) GO TO 90
      IF(ITYPE(JPTR) .NE. 2) GO TO 90
      GO TO 20
   10 IF(JPTR .GT. N) GO TO 90
   12 IF(ITYPE(JPTR) .EQ. 2) GO TO 15
      IF(A(JPTR-1) .NE. DECPT) GO TO 90
      LOGST=JPTR
      CALL LOGCHK
      IF(LOG .EQ. 1) GO TO 90
      JPTR=LOGST
      GO TO 20
   15 IF(JPTR .GT. N) GO TO 90
      GO TO 12
   20 IREAL=1
      IF(JPTR .GT. N) GO TO 35
   22 IF(ITYPE(JPTR) .EQ. 2) GO TO 25
      IF(A(JPTR-1) .EQ. EEE) GO TO 24
      IF(A(JPTR-1) .NE. DEE) GO TO 30
      IDES=1
   24 ELOC=JPTR-2
      GO TO 40
   25 IF(JPTR .GT. N) GO TO 35
      GO TO 22
   30 ELOC=JPTR-2
   32 IRELND=ELOC
      RETURN
   35 IRELND=N
      RETURN
   40 IF(JPTR .GT. N) GO TO 32
      NXT=NEXT(JPTR)
      IF(NXT .EQ. PLUS .OR. NXT .EQ. MINUS) GO TO 45
      JPTR=JPTR-1
      IF(ITYPE(JPTR) .NE. 2) GO TO 32
      IF(JPTR .GT. N) GO TO 35
      GO TO 47
   45 IF(JPTR .GT. N) GO TO 32
      IF(ITYPE(JPTR) .NE. 2) GO TO 32
   47 IF(ITYPE(JPTR) .EQ. 2) GO TO 50
      IRELND=JPTR-2
      RETURN
   50 IF(JPTR .GT. N) GO TO 35
      GO TO 47
   90 IREAL=0
      RETURN
      END
      SUBROUTINE RECOG(FIN)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     *LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(1)/LVVSEQ/LVSIZE,LVSQSP
      COMMON/HL/HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON/NEED/START,ASSOC,LEVEL,STOP
      COMMON/STRING/NNN(2),STR
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      INTEGER HOL
      INTEGER START,ASSOC,STOP,RETRN,R,STJ,STACK,STR(1),ACTION,STRING,
     * RTEMP
      LOGICAL FAIL,FIN
      FIN=.FALSE.
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      JSTACK=0
      J=1
C     START "R
C****      START           "           R
             R =      START
C     STRING+STRING.J/70 "STJ
C****     STRING           +      STRING
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=     STRING
      LVVARG=     STRING
      CALL LVFIND(LV2      A,LV2      B,LV2      C,LV2      D)
      LV1    AAD =    STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         70
C**** LV1     AAD          "         STJ
           STJ = LV1    AAD
      M=-1
C     STRING+HOL.1 STRING ""M""
C****     STRING           +          HOL
      LVVPOS =          1
      LVVTYP=3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2      E,LV2      F,LV2      G,LV2      H)
      LV1    AAD =    STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1     AAD     STRING          ""
      LVDEST=  0
      LV1    AAE = M
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAE
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1    AAD
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
    6 CONTINUE
C   10 R(+ASSOC//15,+STOP//15,=STOP/20)
   10 CONTINUE
      LV1    AAD =           R
C**** LV1    AAD           +       ASSOC
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       ASSOC
      LVVARG= LV1    AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAF = LV1    AAD
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          15
C**** LV1    AAD           +        STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STOP
      LVVARG= LV1    AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAF = LV1    AAD
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          15
      LVVAL = -100
      IF (LV1    AAD.NE.       STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          20
   15 JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C   20 R+ACTION/22 "N
   20 CONTINUE
C****          R            +      ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=      ACTION
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =           R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          22
C**** LV1    AAD           "           N
```

```
               N = LV1    AAD
      CALL SEMANT(N,FAIL)
      IF(FAIL) GO TO 99
      GO TO 25
C  22 R+STJ/99 "R
   22 CONTINUE
C****          R           +        STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX=0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO        99
C**** LV1    AAD           "           R
               R = LV1    AAD
   25 J=J+1
      IF(J .GT. MAXJ) MAXJ=J
C  30 STRING+STRING.J "STJ//6
   30 CONTINUE
C****     STRING           +     STRING
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=     STRING
      LVVARG=     STRING
      CALL LVFIND(LV2      I,LV2     J,LV2      K,LV2      L)
      LV1    AAD =     STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD           "        STJ
             STJ = LV1    AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO         6
   40 STJ=-1
C     R+ACTION/42 "N
C****          R           +     ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=     ACTION
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO        42
C**** LV1    AAD           "           N
               N = LV1    AAD
      CALL SEMANT(N,FAIL)
      IF(FAIL) GO TO 99
   42 CALL SSTOP(FAIL)
      IF(FAIL) GO TO 99
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C     R+ACTION/44 "N
C****          R           +      ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=     ACTION
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         44
C**** LV1    AAD           "           N
               N = LV1    AAD
      CALL SEMANT(N,FAIL)
      IF(FAIL) GO TO 99
   44 CALL SLEVEL(FAIL)
      IF(FAIL) RETURN
      GO TO 40
   99 CONTINUE
      CALL RECOV(RETRN)
      IF(RETRN .LT. 0) GO TO 70
C     RETRN=ASSOC//30
C****    RETRN             =       ASSOC
      LVVAL = -100
      IF (     RETRN.NE.     ASSOC) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO         30
      IF(RETRN .EQ. 0) GO TO 10
      CALL SLEVEL(FAIL)
      IF(FAIL) GO TO 65
      GO TO 30
   65 IF(JSTACK .LE. 1) GO TO 70
      JSTACK=JSTACK-1
      GO TO 99
   70 FIN=.TRUE.
      RETURN
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      LV2I=0
      LV2J=0
      LV2K=0
      LV2L=0
      GO TO 25001
      END
      SUBROUTINE RECOV(RETRN)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      INTEGER START,ASSOC,STOP,STACK,STR(1),STJ,R,STRING,RETRN,TEMP
     +,RIGHT,HOL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
   10 R=STACK(JSTACK,1)
      JAS=STACK(JSTACK,2)+1
C     R+ASSOC.JAS "TEMP//30
C****          R           +       ASSOC
      LVVPOS =        JAS
      LVVTYP =   3
      LVFUNC=      ASSOC
      LVVARG=          R
      CALL LVFIND(LV2     A,LV2      B,LV2      C,LV2      D)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD           "        TEMP
            TEMP = LV1    AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO         30
C   15 R(=STOP//40.+STOP/16=STOP//40 "R)
   15 CONTINUE
      LV1    AAD =          R
C**** LV1    AAD           =        STOP
      LVVAL = -100
      IF (LV1    AAD.NE.      STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO         40
C**** LV1    AAD           +        STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       STOP
      LVVARG= LV1    AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAF = LV1    AAD
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         16
C**** LV1    AAF           =        STOP
      LVVAL = -100
      IF (LV1    AAF.NE.      STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          40
C**** LV1    AAF             "             R
               R = LV1    AAF
      J=STACK(JSTACK,3)
      JSTACK=JSTACK-1
      RETRN=0
      RETURN
   16 JSTACK=JSTACK-1
      IF(JSTACK .LE. 0) GO TO 20
      IF(STACK(JSTACK,3) .LT. J) CALL SEMANT(0,FAIL)
      J=STACK(JSTACK,3)
      GO TO 10
   20 RETRN=-1
      RETURN
   40 CONTINUE
      J=STACK(JSTACK,3)
      IF(STACK(JSTACK,4) .GT. 0) GO TO 16
      RETRN=STOP
      RETURN
C  30 TEMP "R
   30 CONTINUE
C****       TEMP             "             R
               R =        TEMP
      IF(JSTACK .EQ. 1) GO TO 35
      IF( R .NE. STACK(JSTACK,1)) GO TO 35
      NTEMP=STACK(JSTACK-1,1)
      JMARK=JSTACK
   31 STACK(JMARK,4)=-1
      JMARK=JMARK-1
      IF(R .EQ. STACK(JMARK,1) .AND. STACK(JMARK,4) .LT. 0) GO TO 15
      IF(R .EQ. STACK(JMARK,1) .AND. JMARK .NE. 0) GO TO 31
      IF(R .NE. NTEMP .OR. JAS .NE. STACK(JSTACK-1,2) .OR.
     + STACK(JSTACK-1,4) .GE. 0) GO TO 35
      GO TO 15
   35 CONTINUE
      IF(STACK(JSTACK,3) .LT. J) CALL SEMANT(0,FAIL)
      STACK(JSTACK,2)=JAS
      J=STACK(JSTACK,3)
      IF(STACK(JSTACK,4) .GT. 0) STACK(JSTACK,4)=0
      RETRN=ASSOC
      RETURN
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      GO TO 25001
      END
      SUBROUTINE ROLCHK(I1,I2,I3,I4,I5,I6)
      DIMENSION ARG(6),SUBNM(2),ISUBNM(2)
      EQUIVALENCE (ISUBNM(1),SUBNM(1))
      DATA MASK/ZFF000000/
      ARG(1)=AND(I1,MASK)
      ARG(2)=AND(I2,MASK)
      ARG(3)=AND(I3,MASK)
      ARG(4)=AND(I4,MASK)
      ARG(5)=AND(I5,MASK)
      ARG(6)=AND(I6,MASK)
      ISUBNM(1)=0
      ISUBNM(2)=0
      NCHAR=0
      J=0
   10 J=J+1
      DO 30 I=1,4
      NCHAR=NCHAR+1
      IF(I .EQ. 1) GO TO 20
      NSHIFT=8*(I-1)
      CALL SHIFTR(ARG(NCHAR),NSHIFT)
   20 SUBNM(J)=OR(SUBNM(J),ARG(NCHAR))
      IF(NCHAR .EQ. 6) GO TO 35
   30 CONTINUE
      GO TO 10
   35 WRITE(3,40) ISUBNM(1),ISUBNM(2)
   40 FORMAT(A4,A2)
      RETURN
      END
```

```
      SUBROUTINE SEARCH
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DO 20 IDTYP=1,2
      J=INITID(IDTYP)
      IF(J .EQ. 0) GO TO 15
      DO 10 I=1,NID
      IF(ICOMP(NXTID,IDTBL,J,11) .EQ. 0) GO TO 5
      ISRCH(IDTYP)=1
      IDES=LOC
      LOC=J
      GO TO 20
    5 J=IDTBL(4,J)
      IF(J .EQ. 0) GO TO 15
   10 CONTINUE
   15 ISRCH(IDTYP)=0
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SEMANT(N.FAIL)
      COMMON/LVARGS/LVFUNC.LVVARG.LVVAD.LVVPOS.LVVTYP.        LVVAL.
     +LVHEAD.LVVNVL.LVDEST.LVVALS(10).LVTYPE(10).LVSKIP
      COMMON/LVTABL/LVTSIZ.LVMAP(     1)/LVVSEG/LVSIZE.LVSQSP(      1)
      COMMON/FUNC/  NARY(5.22).MARGS.IARGS(50).FNCLOC(5).NFUNC
      COMMON/HL/HOL.ACTION.FUNC1.FUNC2.FUNC3.LEFT.RIGHT.STRING.MAXJ
      COMMON /TYP/ NARRAY.TYPE1.TYPE2.ERRFLG
      COMMON /STRING/ NTYPE.NSTR.STR
      COMMON /JL/ JSTOP
      COMMON /GIRL/NTERMS.PLUS.MINUS.SLASH,LPAR,RPAR,COMMA,STAR.EXP.LT,
     +LE,GT,GE,EQ.NE.OR.AND,NOT,EQUALS,OPRAND
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /NEED/ START.ASSOC.LEVEL,STOP
      COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG
      INTEGER HOL.ACTION,FUNC.LEFT,RIGHT.STRING,RPAR.STJ.R,STACK
     $ ,EXP.FUNC1.FUNC2.FUNC3.TYPE1.TYPE2.TYPE(2,5).STR(1).STOP
     $ ,ALPHA,BETA,GAMMA,OPRAND,EQUALS,AND,OR,COMMA
      LOGICAL SKIP.FLAG,ERRFLG,FAIL,NOTFLG
      INTEGER FUNCRF.BITPUT,PLUS,FL(3).BITGET
      INTEGER GETTYP.GETDIM
      DATA FLAG/.FALSE./.FUNCRF/86/
      DATA TYPE/4HREAL,1H ,4HCOMP,3HLEX,4HDOUB,
     * 2HLE.4HINTE.3HGER,4HLOGI,3HCAL/
      GETTYP(II)=MOD(II,100000)/10000
      GETDIM(II)=MOD(II,1000000)/100000
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      FAIL=.FALSE.
      IF(N .EQ. 0) GO TO 999
      GO TO(10,20,30,40,50,60,70,80,90,1000,1100,1200,1300,1400,1500,
     $ 1600),N
C  10 R+STJ/11 "R//12
   10 CONTINUE
C****          R             +            STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =             R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO            11
C**** LV1    AAD              "            R
               R = LV1    AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO            12
   11 FAIL=.TRUE.
      RETURN
C PRIMARY RECOGNIZED
   12 IF(STJ .EQ. PLUS .OR. STJ .EQ. MINUS) GO TO 126
```

```
      IF(STJ .NE. RPAR) GO TO 121
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=STOP
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
      NTMP=R
      CALL SLEVEL(SKIP)
      JSTACK=JSTACK-1
      R=NTMP
      JLAST=1
      IF(JSTOP .GT. 0) JLAST=STACK(JSTOP,3)
C     STRING+HOL.JLAST(-STRING.STRING ""TYPE1"")
C****     STRING            +          HOL
      LVVPOS =       JLAST
      LVVTYP =   3
      LVFUNC=          HOL
      LVVARG=      STRING
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LV1    AAD =      STRING
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
      LV1    AAF = LV1     AAD
C**** LV1    AAF            -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=LV1     AAF
      CALL LVDLET
      LV1    AAF = LV1     AAD
C**** LV1    AAF       STRING            ""
      LVDEST=  0
      LV1    AAG = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAG
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1     AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
  121 CONTINUE
C  GET TYPE
      BETA=GETDIM(STR(J))
      IF(BETA .NE. 5) GO TO 125
C  OPERAND IS A FUNCTION REFERENCE
      IF(NDEP .EQ. 0) GO TO 18
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=          1
      LVDEST=  2
      LV1    AAD = 1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAD
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =       FUNC1
      LVVARG=    OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
   18 R=FUNCRF
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J+1
      STACK(JSTACK,4)=0
  125 ALPHA=GETTYP(STR(J))
      IF(TYPE1 .GE. 0) GO TO 13
C  SET TYPE OF STATEMENT
      TYPE1=ALPHA
      IF(NTYPE .EQ. 3) TYPE1=-1
C 126 STRING+HOL.J(-STRING,STRING ""TYPE1"")
  126 CONTINUE
C****     STRING           +          HOL
      LVVPOS =            J
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2      E,LV2      F,LV2      G,LV2      H)
      LV1    AAD =     STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LV1    AAF = LV1    AAD
C**** LV1    AAF           -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAF
      CALL LVULET
      LV1    AAF = LV1    AAD
C**** LV1    AAF      STRING          ""
      LVDEST=  0
      LV1    AAH = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAH
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1    AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
   13 IF(FLAG) GO TO 15
C  CHECK FOR MIXED MODE EXPRESSION
      IF(TYPE1 .EQ. ALPHA .OR. ALPHA .EQ. 5) GO TO 16
   14 N1=TYPE1+1
```

```
      N2=ALPHA+1
      ERRFLG=.TRUE.
      CALL ERROR(77,TYPE(1,N1),TYPE(2,N1),TYPE(1,N2),TYPE(2,N2))
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****     STRING              +          HOL
      LVVPOS =             J
      LVVTYP =   3
      LVFUNC=          HOL
      LVVARG=     STRING
      CALL LVFIND(LV2       I,LV2       J,LV2       K,LV2       L)
      LV1    AAD =      STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LV1    AAF = LV1    AAD
C**** LV1    AAF              -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAF
      CALL LVDLET
      LV1    AAF = LV1    AAD
C**** LV1    AAF       STRING           ""
      LVDEST=  0
      LV1    AAI = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAI
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1    AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
C  PARSING  AN EXPONENT
   15 IF((ALPHA .EQ. 3 .AND. TYPE1 .EQ. 3).OR.((TYPE1 .EQ. 0 .OR. TYPE1
     + .EQ. 2) .AND.
     +(ALPHA .EQ. 0 .OR. ALPHA .EQ. 2))) GO TO 16
      CALL ERROR(78,J,IDM2,IDM3,IDM4)
      ERRFLG=.TRUE.
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****     STRING              +          HOL
      LVVPOS =             J
      LVVTYP =   3
      LVFUNC=          HOL
      LVVARG=     STRING
      CALL LVFIND(LV2       M,LV2       N,LV2       O,LV2       P)
      LV1    AAD =      STRING
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LV1    AAF = LV1    AAD
C**** LV1    AAF              -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1     AAF
      CALL LVDLET
      LV1     AAF = LV1     AAD
C**** LV1     AAF        STRING            ""
      LVDEST=  0
      LV1     AAJ = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAJ
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1     AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
   16 IF((.NOT. FLAG .AND. TYPE1 .LT. ALPHA).OR.(FLAG .AND. ALPHA .NE. 3
     + .AND. TYPE1 .LT. ALPHA)) TYPE1=ALPHA
C     STRING+HOL.J(-STRING.STRING ""TYPE1"")
C****      STRING              +           HOL
      LVVPOS =              J
      LVVTYP =    3
      LVFUNC=        HOL
      LVVARG=      STRING
      CALL LVFIND(LV2       Q.LV2       R.LV2       S.LV2       T)
      LV1     AAD =      STRING
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
      LV1     AAF = LV1     AAD
C**** LV1     AAF             -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1     AAF
      CALL LVDLET
      LV1     AAF = LV1     AAD
C**** LV1     AAF        STRING            ""
      LVDEST=  0
      LV1     AAK = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAK
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1     AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
C  WILL SCAN AN EXPONENT
   20 IF(STJ .LT. 0) RETURN
C     R+STJ/11 "R
C****          R             +          STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LV1    AAD         "          R
             R = LV1    AAD
      FLAG=.TRUE.
      RETURN
C  RECOGNIZED A TERM,PRODUCT OR PRIMARY PERHAPS NEEDING PARENTHESIZATION
   30 CONTINUE
      KTMP=R
      IF(NDEP .EQ. 0) GO TO 34
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=          1
      LVDEST=  2
      LV1    AAH = 1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAH
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =      FUNC1
      LVVARG=    OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
   34 CONTINUE
      ITEST=0
      IF(STJ .LT. 0) GO TO 31
C     R+STJ "R//32
C****         R          +        STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAF =          R
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL
C**** LV1    AAF         "          R
             R = LV1    AAF
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          32
C  31 R+STOP/39 "R
   31 CONTINUE
C****         R          +        STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       STOP                                                00003310
      LVVARG=          R                                                00003320
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)                          00003330
      LV1    AAF =          R                                           00003340
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL                               00003350
      LVVTR = LVVAL                                                     00003360
      LVVAL = -100                                                      00003370
      IF (LVVTR.EQ.-1) GO TO         39                                 00003380
C**** LV1    AAF             "            R                             00003390
             R = LV1    AAF                                             00003400
      IF(STJ .LT. 0 .AND. KTMP .EQ. 889) GO TO 38                       00003410
      ITEST=-1                                                          00003420
   32 CONTINUE                                                          00003430
C IF UNARY PLUS OR MINUS RETURN                                         00003440
      IF(STACK(JSTOP,1) .NE. 288 .AND. STACK(JSTOP,1) .NE. 110) GO TO 3300003450
      JLAST=STACK(JSTOP,3)-1                                            00003460
      IF(STACK(JSTOP,1) .EQ. 288) JLAST=JLAST-1                         00003470
C     STRING+HOL.JLAST(-STRING,STRING ""TYPE1"")                        00003480
C****     STRING             +          HOL                             00003490
      LVVPOS =      JLAST                                               00003500
      LVVTYP =   3                                                      00003510
      LVFUNC=       HOL                                                 00003520
      LVVARG=     STRING                                                00003530
      CALL LVFIND(LV2      U,LV2      V,LV2      W,LV2      X)          00003540
      LV1    AAF =     STRING                                           00003550
      IF (LVVAL.NE.-1) LV1    AAF = LVVAL                               00003560
      LV1    AAL = LV1    AAF                                           00003570
C**** LV1    AAL             -       STRING                             00003580
      LVVAD=-1                                                          00003590
      LVVTYP=-1                                                         00003600
      LVVPOS=1                                                          00003610
      LVFUNC=    STRING                                                 00003620
      LVVARG=LV1    AAL                                                 00003630
      CALL LVDLET                                                       00003640
      LV1    AAL = LV1    AAF                                           00003650
C**** LV1    AAL      STRING            ""                              00003660
      LVDEST=  0                                                        00003670
      LV1    AAM = TYPE1                                                00003680
      LVTYPE(1) =  1                                                    00003690
      LVVALS(1) = LV1    AAM                                            00003700
      LVDEST=  0                                                        00003710
      LVVNVL =  1                                                       00003720
      LVFUNC =      STRING                                              00003730
      LVVARG=LV1    AAL                                                 00003740
      CALL LVNSRT                                                       00003750
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)                                 00003760
      IF(LVVAL.LT.0) RETURN                                             00003770
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")                            00003780
C****     STRING             +          HOL                             00003790
      LVVPOS =           J                                              00003800
      LVVTYP =   3                                                      00003810
      LVFUNC=       HOL                                                 00003820
      LVVARG=     STRING                                                00003830
      CALL LVFIND(LV2      Y,LV2      Z,LV2      0,LV2      1)          00003840
      LV1    AAF =     STRING                                           00003850
      IF (LVVAL.NE.-1) LV1     AAF = LVVAL
      LV1     AAL = LV1     AAF
C**** LV1     AAL              -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=LV1     AAL
      CALL LVDLET
      LV1     AAL = LV1     AAF
C**** LV1     AAL       STRING           ""
      LVDEST=  0
      LV1     AAN = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAN
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1     AAL
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(ITEST .LT. 0) J=J-1
      RETURN
   33 CONTINUE
      IF(ITEST .LT. 0) J=J-1
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=STOP
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
      NTMP=R
      CALL SLEVEL(SKIP)
      JSTACK=JSTACK-1
      R=NTMP
      JLAST=1
      IF(JSTOP .GT. 0) JLAST=STACK(JSTOP,3)
      NTMP=TYPE1
      JJ=JLAST
      IF(STACK(JSTACK,1) .EQ. 418 .AND. JLAST .GT. 1) JJ=JLAST-1
C     STRING+HOL.JJ+STRING "TYPE1
C****      STRING              +           HOL
      LVVPOS =           JJ
      LVVTYP =   3
      LVFUNC=          HOL
      LVVARG=      STRING
      CALL LVFIND(LV2       2,LV2       3,LV2       4,LV2       5)
      LV1     AAF =      STRING
      IF (LVVAL.NE.-1) LV1     AAF = LVVAL
C**** LV1     AAF              +       STRING
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=      STRING
      LVVARG= LV1     AAF
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAL = LV1    AAF
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
C**** LV1    AAL          "       TYPE1
         TYPE1 = LV1    AAL
      IF(.NOT. FLAG .OR. NTMP .EQ. 3) GO TO 35
      IF((NTMP.EQ.0.OR.NTMP.EQ.2).AND.(TYPE1.EQ.0.OR.TYPE1.EQ.2))GOTO 35
      ERRFLG=.TRUE.
      CALL ERROR(78,J,IDM2,IDM3,IDM4)
   35 CONTINUE
      IF(TYPE1 .GT. 2 .OR. NTYPE .GT. 1) GO TO 38
      FUNC=FUNC1
      IF(TYPE1 .EQ. 1) FUNC=FUNC2
      IF(TYPE1 .EQ. 2) FUNC=FUNC3
C     STRING+HOL.JLAST LEFT(.1 LPAR,.1 FUNC)
C****     STRING          +         HOL
      LVVPOS =      JLAST
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2      6,LV2      7,LV2      8,LV2      9)
      LV1    AAL =     STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LV1    AAF = LV1    AAL
      LV1    AAO =       LEFT
C**** LV1    AAO          .         1
      LVVTYP=  3
      LVVPOS=        1
C**** LV1    AAF  LV1    AAO        LPAR
      LVDEST=  1
      LVTYPE(1) =  0
      LVVALS(1) =       LPAR
      LVVNVL =  1
      LVFUNC = LV1    AAO
      LVVARG=LV1    AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LV1    AAF = LV1    AAL
C**** LV1    AAO          .         1
      LVVTYP=  3
      LVVPOS=        1
C**** LV1    AAF  LV1    AAO        FUNC
      LVDEST=  1
      LVTYPE(1) =  0
      LVVALS(1) =       FUNC
      LVVNVL =  1
      LVFUNC = LV1    AAO
      LVVARG=LV1    AAF
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C     STRING+HOL.J RIGHT RPAR
C****     STRING          +         HOL
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=     STRING
      CALL LVFIND(LV2     AA,LV2     AB,LV2     AC,LV2     AD)
      LV1    AAL =     STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
C**** LV1    AAL        RIGHT         RPAR
      LVDEST=  0
      LVTYPE(1) =  0
      LVVALS(1) =       RPAR
      LVVNVL =  1
      LVFUNC =       RIGHT
      LVVARG=LV1    AAL
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
   38 IF(ITEST .LT. 0 .AND. STJ .LT. 0) J=J+1
      FLAG=.FALSE.
      RETURN
   39 FLAG=.FALSE.
      GO TO 11
C  CHECK FOR CORRECTNESS OF SUBSCRIPTS
   40 NR=R
C     R+STJ/11 "R
C****          R            +          STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=         STJ
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAL =           R
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LV1    AAL            "            R
               R = LV1    AAL
      NBETA=GETDIM(STR(J))
      IF(NR .EQ. 359 .AND. NBETA .NE. 4) GO TO 47
      ALPHA=GETTYP(STR(J))
      GAMMA=STR(J)/1000000
      IF(NTYPE .EQ. 3) GO TO 45
      IF(NR .EQ. 839 .AND. NBETA .EQ. 0) GO TO 45
      IF(NR .EQ. 359) GO TO 45
      IF(NR .EQ. 21 .AND. NBETA .EQ. 4) GO TO 45
      IF(NBETA .EQ. 0 .AND. NR .EQ. 935) GO TO 45
      IF(NTYPE .EQ. 2 .AND. NR .EQ. 935 .AND. STR(J-1) .NE. -7) GO TO 45
      IF(NTYPE .EQ. 2 .AND. NR .EQ. 359 .AND. NBETA .EQ. 0) GO TO 11
      CALL ERROR(79,J,IDM2,IDM3,IDM4)
      ERRFLG=.TRUE.
   45 CONTINUE
      IF(GAMMA .GE. 6 .AND. NBETA .EQ. 4)
     $ CALL ERROR(76,IDM1,IDM2,IDM3,IDM4)
      IF(ALPHA .EQ. 3) GO TO 46
      N1=ALPHA+1
      ERRFLG=.TRUE.
      CALL ERROR(80,TYPE(1,N1),TYPE(2,N1),J,IDM4)
   46 IF(NBETA .EQ. 4) RETURN
      MARGS=MARGS+1
      LV1    AAO =     OPRAND
      LVDEST=  0
      LV1    AAR = MARGS
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAR
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC2
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1    AAJ = J
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAJ
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC3
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1    AAS = NDEPTH
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAS
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       LEVEL
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IVR=MARGS
      IF(IVR .GT. 50) GO TO 1610
      ICOL=9
      IVAL=MOD(STR(J),10000)
      IARGS(IVR)=BITPUT(IARGS(IVR),IVAL,ICOL)
      IF(NR .EQ. 839) GO TO 49
      IF(NTYPE .NE. 3) RETURN
C  FLAG SUBSCRIPT IN I/O LIST
      IARGS(IVR)=BITPUT(IARGS(IVR),1,ICOL+2)
      RETURN
C FLAG DO INDEX IN I/O LIST
   49 IARGS(IVR)=BITPUT(IARGS(IVR),2,ICOL+2)
      IF(NFLAG .LT. 1) RETURN
      IARGS(IVR)=BITPUT(IARGS(IVR),FL(NFLAG),ICOL+7)
      RETURN
   47 NR=R
```

```
C    SUBSCRIPT DOES NOT BEGIN WITH CONSTANT, FORCE SEARCH FOR VARIABLE
        GO TO 11
C   CHECK FOR PROPER NUMBER OF SUBSCRIPTS
   50 IF( BETA .EQ. 4 .OR. R .NE. 452) GO TO 52
      MARGS=MARGS+1
      LV1     AAO =      OPRAND
      LVDEST=  0
      LV      AAT = MARGS
      L  YPE(1) =  1
      LVVALS(1) = LV1     AAT
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC2
      LVVARG=LV1     AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1     AAU = J-1
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAU
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC3
      LVVARG=LV1     AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      LVDEST=  0
      LV1     AAV = NDEPTH
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAV
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       LEVEL
      LVVARG=LV1     AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(MARGS .GT. 50) GO TO 1610
      ICOL=9
      IVAL=MOD(STR(J-1),10000)
      IARGS(MARGS)=BITPUT(IARGS(MARGS),IVAL,ICOL)
      IF(NOPAR .LE. 0) GO TO 52
      LVVPOS =          1
      LVVTYP =    3
      LVVPOS=-LVVPOS
      LVFUNC=     ACTION
      LVVARG=     OPRAND
      CALL LVFIND(LV2     AE,LV2      AF,LV2      AG,LV2      AH)
      LV1     AAO =      OPRAND
      IF (LVVAL.NE.-1) LV1     AAO = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         52
```

```
          MFUNC = LV1    AAO
      IARGS(MARGS)=HITPUT(IARGS(MARGS),MFUNC,ICOL+3)
      IAHGS(MARGS)=BITPUT(IAHGS(MARGS),NARGS,ICOL+9)
C IF NO STHING LEFT, RETURN IF CONSTANT,VARIAHLE OR I/O LIST
   52 IF(STJ .LT. 0 .AND. (BETA .EQ. 0 .OR. HETA .EQ. 4
     $ .OR. NTYPE .EQ. 3)) RETURN
      IF(HETA .GT. NARRAY) GO TO 55
      ERRFLG=.TRUE.
      CALL ERROR(81,J,IDM2,IDM3,IDM4)
   55 IF(R .EQ. 452) NARRAY=0
      IF(R .EQ. 318) NARRAY=1
      IF(R .EQ.  60) NARRAY=2
      IF(R .EQ. 103) NARRAY=3
      IF(STJ .LT. 0) GO TO 58
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=         STJ
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAL =            R
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
C**** LV1    AAL          "           R
             R = LV1    AAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          56
   58 IF(NTYPE .EQ. 3 .AND. NARRAY .EQ. 0) GO TO 57
      IF(HETA .GE. 1 .AND. HETA .LE. 3 .AND. NOPAR .EQ. 0)
     $ CALL ERROR(82,J,IDM2,IDM3,IDM4)
   57 NARRAY=-1
      GO TO 11
   56 IF(NTYPE .EQ. 3 .AND. NARRAY .EQ. 0) RETURN
      IF(STJ .EQ. RPAR .AND. NARRAY .LT.  BETA .AND. J .EQ. MAXJ)
     $ CALL ERROR(82,J,IDM2,IDM3,IDM4)
      IF(STJ .EQ. RPAR) NARRAY=-1
      RETURN
C  RESET TYPE OF STATEMENT IN ANTICIPATION OF SEARCH FOR BOOLEAN PRIMARY
   60 CONTINUE
      NOTFLG=.FALSE.
      IF(STR(J-1) .EQ. -17) NOTFLG=.TRUE.
      TYPE1=-1
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****     STRING          +         HOL
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=     STRING
      CALL LVFIND(LV2AI,LV2AJ,LV2AK,LV2AL)
      LV1    AAL =      STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LV1    AAO = LV1    AAL
C**** LV1    AAO          -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1     AAO
      CALL LVDLET
      LV1    AAO = LV1    AAL
C**** LV1    AAO        STRING            ""
      LVDEST=  0
      LV1    AAF = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAF
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       STRING
      LVVARG=LV1     AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(STJ .NE. OPRAND) GO TO 65
      ALPHA=GETTYP(STR(J))
      BETA=GETDIM(STR(J))
      IF(ALPHA .NE. 4) GO TO 11
   65 CONTINUE
C     R+STJ/11 "R
C****          R          +          STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=         STJ
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAL =           R
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LV1    AAL          "          R
               R = LV1    AAL
      RETURN
C  IF BOOLEAN PRIMARY IS AN ARETHMETIC COMPARE CONTINUE PARSING PRIMARY
   70 IF(STJ .LT. 0) RETURN
      IF(TYPE1 .EQ. 4) GO TO 75
C     R+STJ "R/11
C****          R          +          STJ
      LVVTYP =' 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=         STJ
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAL =           R
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
C**** LV1    AAL          "          R
               R = LV1    AAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C  RELATIONAL OPERATOR FOUND
      IF(NDEP .EQ. 0) RETURN
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=           1
      LVDEST=  2
      LV1    AAO = 1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAO
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =      FUNC1
      LVVARG=    OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C  IF BOOLEAN VARIABLE OR CONSTANT, SET STATE TO STOP
   75 R=STOP
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
      GO TO 11
C  COMPARE TYPES ON BOTH SIDES OF RELATIONAL EXPRESSION
   80 IF(TYPE1 .EQ. 0 .OR. TYPE1 .EQ. 2 .OR. TYPE1 .EQ. 3) GO TO 85
      ERRFLG=.TRUE.
      CALL ERROR(83,J)
      TYPE1=-1
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****      STRING            +           HOL
      LVVPOS =           J
      LVVTYP =   3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2AM,LV2AN,LV2AO,LV2AP)
      LV1    AAL =      STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LV1    AAO = LV1    AAL
C**** LV1    AAO            -      STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAO
      CALL LVDLET
      LV1    AAO = LV1    AAL
C**** LV1    AAO       STRING          ""
      LVDEST=  0
      LV1    AAP = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAP
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
   85 TYPE2=TYPE1
      GO TO 11
C  BOOLEAN PRIMARY RECOGNIZED-SET TYPE TO BOOLEAN AND CONTINUE PARSE
   90 IF(TYPE1 .EQ. TYPE2 .OR. TYPE1+TYPE2 .EQ. 2 .OR.
     + TYPE2 .LT. 0) GO TO 95
      N1=TYPE1+1
      N2=TYPE2+1
      CALL ERROR(77,TYPE(1,N1),TYPE(2,N1),TYPE(1,N2),TYPE(2,N2))
      ERRFLG=.TRUE.
   95 TYPE1=4
      TYPE2=-1
      IF(STJ .LT. 0) RETURN
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****     STRING             +          HOL
      LVVPOS =          J
      LVVTYP =    3
      LVFUNC=         HOL
      LVVARG=     STRING
      CALL LVFIND(LV2AQ,LV2AR,LV2AS,LV2AT)
      LV1    AAL =      STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LV1    AAO = LV1    AAL
C**** LV1    AAO             -        STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAO
      CALL LVDLET
      LV1    AAO = LV1    AAL
C**** LV1    AAO       STRING              ""
      LVDEST=  0
      LV1    AAQ = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAQ
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      GO TO 11
C  PARSE REACHED BLIND ALLEY-MUST BACK UP AND REMOVE PARENTHESES CREATED
  999 JM=STACK(JSTACK,3)
      K=JM
      DO 996 KK=JM,J
C     STRING+HOL.K/995+STRING.1/996 "TYPE1
C****     STRING             +          HOL
      LVVPOS =          K
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=      STRING
      CALL LVFIND(LV2AU,LV2AV,LV2AW,LV2AX)
      LV1    AAL =      STRING
      IF (LVVAL.NE.-1) LV1    AAL = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         995
C**** LV1    AAL              +       STRING
      LVVPOS =             1
      LVVTYP =   3
      LVFUNC=      STRING
      LVVARG= LV1    AAL
      CALL LVFIND(LV2AY,LV2AZ,LV2A0,LV2A1)
      LV1    AAO = LV1    AAL
      IF (LVVAL.NE.-1) LV1    AAO = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO         996
C**** LV1    AAO              "       TYPE1
         TYPE1 = LV1    AAO
      GO TO 995
  996 K=K-1
  995 CONTINUE
      DO 998 I=JM,J
C     STRING+HOL.I(-LEFT,-RIGHT)
C****    STRING               +          HOL
      LVVPOS =             1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=      STRING
      CALL LVFIND(LV2A2,LV2A3,LV2A4,LV2A5)
      LV1    AAO =      STRING
      IF (LVVAL.NE.-1) LV1    AAO = LVVAL
      LV1    AAL = LV1    AAO
C**** LV1    AAL              -          LEFT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=      LEFT
      LVVARG=LV1    AAL
      CALL LVDLET
      LV1    AAL = LV1    AAO
C**** LV1    AAL              -        RIGHT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     RIGHT
      LVVARG=LV1    AAL
      CALL LVDLET
  998 CONTINUE
  980 CONTINUE
      LVVPOS =             1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC3
      LVVARG=     OPRAND
      CALL LVFIND(LV2     A6,LV2     A7,LV2     A8,LV2     A9)
      LV1    AAY =     OPRAND
      IF (LVVAL.NE.-1) LV1    AAY = LVVAL
           JN = LV1    AAY
      IF(JN .LT. JM) GO TO 985
      LV1    AAY =     OPRAND
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC2
      LVVARG= LV1    AAY
      LVVAD=-1
      CALL LVDLET
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC3
      LVVARG= LV1    AAY
      LVVAD=-1
      CALL LVDLET
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      LEVEL
      LVVARG= LV1    AAY
      LVVAD=-1
      CALL LVDLET
      GO TO 980
  985 CONTINUE
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC2
      LVVARG=     OPRAND
      CALL LVFIND(LV2     BA,LV2     BB,LV2     BC,LV2     BD)
      LV1    AAY =     OPRAND
      IF (LVVAL.NE.-1) LV1    AAY = LVVAL
         MARGS = LV1    AAY
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      LEVEL
      LVVARG=     OPRAND
      CALL LVFIND(LV2     BE,LV2     BF,LV2     BG,LV2     BH)
      LV1    AAY =     OPRAND
      IF (LVVAL.NE.-1) LV1    AAY = LVVAL
        NDEPTH = LV1    AAY
      RETURN

C   RECOGNIZED FUNCTION-PREPARE TO SET TYPE OF ARGUMENTS FOR THE "NDEPTH
C           FUNCTION IN THIS STMT
 1000 CONTINUE
```

```
      NDEPTH=NDEPTH+1
      NDEP=NDEP+1
      NARGS=0
C     R+STJ/11 "R
C****         R            +           STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       STJ
      LVVARG=         R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAO =           R
      IF (LVVAL.NE.-1) LV1    AAO = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LV1    AAO           "            R
               R = LV1    AAO
      NARGS=1
C     OPRAND(OPRAND ""TYPE1"",STRING""NARGS"",ACTION ""NDEPTH"")
      LV1    AAO =     OPRAND
C**** LV1    AAO      OPRAND          ""
      LVDEST=  0
      LV1    AAL = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAL
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     OPRAND
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C**** LV1    AAO      STRING          ""
      LVDEST=  0
      LV1    AAR = NARGS
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAR
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C**** LV1    AAO      ACTION          ""
      LVDEST=  0
      LV1    AAS = NDEPTH
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAS
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     ACTION
      LVVARG=LV1    AAO
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C     OPRAND FUNC1 ""J+1""
C****     OPRAND        FUNC1           ""
      LVDEST=  0
      LV1AAO=0
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAO
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =       FUNC1
      LVVARG=     OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      TYPE1=-1
C     NOPAR=NOPAR+1
      NOPAR=NOPAR+1
      RETURN

C KEEP TRACK OF THE NUMBER AND TYPES OF ARGUMENTS IN FUNCTION CALLS
C   MUST USE STACK FOR POSSIBLE RECURSIVE FUNCTION USE
 1100 CONTINUE
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1AAT=R
      IF(LVVAL .NE. -1) LV1AAT=LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
      R=LV1AAT
      LV1AAT=OPRAND
      LVVPOS =            1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=     STRING
      LVVARG= LV1    AAT
      CALL LVFIND(LV28I,LV28J,LV28K,LV28L)
      LV1    AAU = LV1    AAT
      IF (LVVAL.NE.-1) LV1    AAU = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO        1103
C**** LV1    AAU           "       NARGS
           NARGS = LV1    AAU
C**** LV1    AAT           +      STRING
      LVVPOS =            1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=     STRING
      LVVARG= LV1    AAT
      LVVAD=-1
      CALL LVDLET
C**** LV1    AAT             +        ACTION
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=     ACTION
      LVVARG= LV1    AAT
      CALL LVFIND(LV2BM,LV2BN,LV2BO,LV2BP)
      LV1    AAU = LV1    AAT
      IF (LVVAL.NE.-1) LV1    AAU = LVVAL
      MFUNC=LV1AAU
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC1
      LVVARG=     OPRAND
      CALL LVFIND(LV2     BQ,LV2     BR,LV2     BS,LV2     BT)
      LV1    AA3 =     OPRAND
      IF (LVVAL.NE.-1) LV1    AA3 = LVVAL
            IEXP = LV1    AA3
 1103 CONTINUE
C  STORE ARGUMENT TYPES
      IF(NDEPTH .GT. 5) CALL ERROR(85,IDM1,IDM2,IDM3,IDM4)
      IF(NDEPTH .GT. 5) GO TO 1130
      IF(NARGS .LE. 63) GO TO 1104
      ERRFLG=.TRUE.
      CALL ERROR(84,NDEPTH,IDM2,IDM3,IDM4)
      GO TO 11
 1104 CONTINUE
      MM=(5+NARGS)/3
      ITEMP=NARGS-3*(MM-2)
      ICOL=9*ITEMP-6
      NARY(MFUNC ,MM)=BITPUT(NARY(MFUNC ,MM),TYPE1+1,ICOL)
      IF(STR(J-2) .NE. -6 .AND. STR(J-2) .NE. -4) GO TO 1130
      NDIM=GETDIM(STR(J-1))
      IF(NDIM .GE. 4) GO TO 1130
C  STORE DIMENSIONALITY OF ARGUMENTS
      NARY(MFUNC ,MM)=BITPUT(NARY(MFUNC ,MM),NDIM,ICOL+3)
 1130 CONTINUE
      NARY(MFUNC,MM)=BITPUT(NARY(MFUNC,MM),IEXP,ICOL+6)
      IF(STJ .EQ. COMMA) GO TO 1105
      NARY(MFUNC,1)=NARGS
      LV1    AA3 =     OPRAND
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=     ACTION
      LVVARG= LV1    AA3
      LVVAD=-1
      CALL LVDLET
      LVVPOS =           1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=     OPRAND
      LVVARG= LV1     AA3
      CALL LVFIND(LV2      BU,LV2      BV,LV2      BW,LV2      BX)
      LV1     AA4 = LV1     AA3
      IF (LVVAL.NE.-1) LV1     AA4 = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO        1135
          TYPE1 = LV1     AA4
      LV1     AA3 =      OPRAND
      LVVPOS =             1
      LVVTYP =    3
      LVVPOS=-LVVPOS
      LVFUNC=       FUNC1
      LVVARG= LV1     AA3
      LVVAD=-1
      CALL LVDLET
      LVVPOS =             1
      LVVTYP =    3
      LVVPOS=-LVVPOS
      LVFUNC=      OPRAND
      LVVARG= LV1     AA3
      LVVAD=-1
      CALL LVDLET
 1135 NOPAR=NOPAR-1
      NDEP=NDEP-1
      RETURN
 1105 TYPEI=-I
C     STRING+HOL.J(-STRING,STRING ""TYPE1"")
C****      STRING              +         HOL
      LVVPOS =             J
      LVVTYP=3
      LVFUNC=         HOL
      LVVARG=      STRING
      CALL LVFIND(LV2BY,LV2BZ,LV2B0,LV2B1)
      LV1     AAT =      STRING
      IF (LVVAL.NE.-1) LV1     AAT = LVVAL
      LV1     AAU = LV1     AAT
C**** LV1     AAU              -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=LV1     AAU
      CALL LVDLET
      LV1     AAU = LV1     AAT
C**** LV1     AAU       STRING             ""
      LVDEST=  0
      LV1     AAV = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1     AAV
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =      STRING
      LVVARG=LV1     AAU
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      NARGS=NARGS+1
C     OPRAND(STRING """NARGS""",FUNC1 .-1 """J+1""")
C**** LV1    AAT      STRING          ""
      LVDEST=  0
      LV1    AAU = NARGS
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAU
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C****     FUNC1          .-          1
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=          1
C**** LV1    AAT       FUNC1          ""
      LVDEST=  2
      LV1AAW=0
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAW
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =      FUNC1
      LVVARG=    OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
C  SAVE TYPE OF STAEMENT WHILE PARSING EXPONENT
 1200 CONTINUE
C     STRING+HOL.J(-STRING.STRING """TYPE1""")
C****     STRING          +          HOL
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=       HOL
      LVVARG=     STRING
      CALL LVFIND(LV2B2.LV2B3.LV2B4.LV2B5)
      LV1    AAT =     STRING
      IF (LVVAL.NE.-1) LV1    AAT = LVVAL
      LV1    AAX = LV1    AAT
C**** LV1    AAX           -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1    AAX
      CALL LVDLET
      LV1    AAX = LV1    AAT
C**** LV1    AAX      STRING          ""
      LVDEST=  0
      LV1    AAY = TYPE1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AAY
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LV1    AAX
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      TYPE1=-1
C     R+STJ/11 "R
C****        R             +         STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAT =          R
      IF (LVVAL.NE.-1) LV1    AAT = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LV1    AAT           "           R
               R = LV1    AAT
C     COMPLETE
      RETURN
 1300 CONTINUE
      IF(STJ .LT. 0) RETURN
      IF(STJ .NE. AND .AND. STJ .NE. OR .AND. STJ .NE. NOT) GO TO 11
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STJ
      LVVARG=          R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AA6 =          R
      IF (LVVAL.NE.-1) LV1    AA6 = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
               R = LV1    AA6
      IF(NDEP .EQ. 0) RETURN
C  LOGICAL OPERATOR FOUND
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=          1
      LVDEST=  2
      LV1    AA6 = 1
      LVTYPE(1) =  1
      LVVALS(1) = LV1    AA6
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =     FUNC1
      LVVARG=     OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
 1400 CONTINUE
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=          STJ
      LVVARG=            R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AA7 =            R
      IF (LVVAL.NE.-1) LV1     AA7 = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
              R = LV1    AA7
C  LEFT PAREN FOUND IN I/O LIST
      NFLAG=NFLAG+1
      FL(NFLAG)=MARGS
      RETURN
 1500 CONTINUE
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=          STJ
      LVVARG=            R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AA7 =            R
      IF (LVVAL.NE.-1) LV1     AA7 = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
              R = LV1    AA7
      IF(STJ .EQ. COMMA) RETURN
C  RIGHT PAREN FOUND IN I/O LIST
      NFLAG=NFLAG-1
      RETURN
 1600 CONTINUE
      PRINT 1601
 1601 FORMAT(1X,10H ACTION 16)
      TYPF1=4
      RETURN
 1610 CALL ERROR(95,IDM1,IDM2,IDM3,IDM4)
      STOP
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      LV2I=0                                                            00013720
      LV2J=0                                                            00013730
      LV2K=0                                                            00013740
      LV2L=0                                                            00013750
      LV2M=0                                                            00013760
      LV2N=0                                                            00013770
      LV2O=0                                                            00013780
      LV2P=0                                                            00013790
      LV2Q=0                                                            00013800
      LV2R=0                                                            00013810
      LV2S=0                                                            00013820
      LV2T=0                                                            00013830
      LV2U=0                                                            00013840
      LV2V=0                                                            00013850
      LV2W=0                                                            00013860
      LV2X=0                                                            00013870
      LV2Y=0                                                            00013880
      LV2Z=0                                                            00013890
      LV20=0                                                            00013900
      LV21=0                                                            00013910
      LV22=0                                                            00013920
      LV23=0                                                            00013930
      LV24=0                                                            00013940
      LV25=0                                                            00013950
      LV26=0                                                            00013960
      LV27=0                                                            00013970
      LV28=0                                                            00013980
      LV29=0                                                            00013990
      LV2AA=0                                                           00014000
      LV2AB=0                                                           00014010
      LV2AC=0                                                           00014020
      LV2AD=0                                                           00014030
      LV2AE=0                                                           00014040
      LV2AF=0                                                           00014050
      LV2AG=0                                                           00014060
      LV2AH=0                                                           00014070
      LV2AI=0                                                           00014080
      LV2AJ=0                                                           00014090
      LV2AK=0                                                           00014100
      LV2AL=0                                                           00014110
      LV2AM=0                                                           00014120
      LV2AN=0                                                           00014130
      LV2AO=0                                                           00014140
      LV2AP=0                                                           00014150
      LV2AQ=0                                                           00014160
      LV2AR=0                                                           00014170
      LV2AS=0                                                           00014180
      LV2AT=0                                                           00014190
      LV2AU=0                                                           00014200
      LV2AV=0                                                           00014210
      LV2AW=0                                                           00014220
      LV2AX=0                                                           00014230
      LV2AY=0                                                           00014240
      LV2AZ=0                                                           00014250
      LV2A0=0                                                           00014260
```

```
LV2A1=0
LV2A2=0
LV2A3=0
LV2A4=0
LV2A5=0
LV2A6=0
LV2A7=0
LV2A8=0
LV2A9=0
LV2BA=0
LV2BB=0
LV2BC=0
LV2BD=0
LV2BE=0
LV2BF=0
LV2BG=0
LV2BH=0
LV2BI=0
LV2BJ=0
LV2BK=0
LV2BL=0
LV2BM=0
LV2BN=0
LV2BO=0
LV2BP=0
LV2BQ=0
LV2BR=0
LV2BS=0
LV2BT=0
LV2BU=0
LV2BV=0
LV2BW=0
LV2BX=0
LV2BY=0
LV2BZ=0
LV2B1=0
LV2B2=0
LV2B3=0
LV2B4=0
LV2B5=0
GO TO 25001
END
```

```
      SUBROUTINE SEPAR
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP
      INTEGER A,SEPST,SEPND,DIR,BLANK,SLASH,COMMA
      DATA BLANK/1H /,SLASH/1H//,COMMA/1H,/
      ICOM=0
      ISLASH=0
      DO 20 I=1,N
      JJ=SEPST+DIR*(I-1)
      IF(A(JJ) .EQ. BLANK) GO TO 20
      IF(A(JJ) .EQ. SLASH) GO TO 5
      IF(A(JJ) .EQ. COMMA) GO TO 10
      GO TO 30
    5 CONTINUE
      ISLASH=1
      IF(ICOM .EQ. 1) GO TO 40
      GO TO 20
   10 IF(ISLASH .EQ. 1 .OR. ICOM .EQ. 1) GO TO 40
      ICOM=1
   20 CONTINUE
      GO TO 40
   30 IF(ISLASH .EQ. 0 .AND. ICOM .EQ. 0) GO TO 35
      ISEP=1
      SEPND=JJ-DIR
      RETURN
   35 CONTINUE
      ISEP=0
      SEPND=JJ
      RETURN
   40 ISEP=-1
      RETURN
      END
      SUBROUTINE SIMP
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH1(7),IALPH2(9),IALPH3(5),IALPH4(10)
      DATA IALPH1/1HP,1HE,1HT,1HU,1HR,1HN,1H /
      DATA IALPH2/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE,1H /
      DATA IALPH3/1HS,1HT,1HO,1HP,1H /
      DATA IALPH4/1HB,1HL,1HO,1HC,1HK,1HD,1HA,1HT,1HA,1H /
      IF(ITYP .EQ. 10) GO TO 25
      IF(ITYP .EQ. 7) GOTO 15
      IF(ITYP .EQ. 29) GO TO 35
      DO 10 I=1,7
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50
   10 CONTINUE
      NB=1
      NBRNCH=1
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=999
      RETURN
   15 DO 20 I=1,9
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50
   20 CONTINUE
      RETURN
   25 DO 30 I=1,5
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50
   30 CONTINUE
      NB=1
      NBRNCH=1
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=999
      RETURN
   35 DO 40 I=1,10
      IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 50
   40 CONTINUE
      RETURN
   50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE SLEVEL(FAIL)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(       1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      COMMON /JL/ JSTOP
      INTEGER START,STOP,ASSOC;STACK,STR(500),STJ,R,RTEMP
      LOGICAL FAIL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      RTEMP=0
      JSTOP=JSTACK
   10 IF(JSTOP .EQ. 0) GO TO 40
      NPNTR=STACK(JSTOP,4)
      IF(NPNTR .GT. 0) GO TO 20
      IF(STACK(JSTOP,2) .NE. 0) GO TO 30
      JSTOP=JSTOP-1
      GO TO 10
   20 JSTOP=NPNTR-1
      GO TO 10
   30 STACK(JSTACK,4)=JSTOP
      JAS=STACK(JSTOP,2)
      R=STACK(JSTOP,1)
      RTEMP=R
C     R+LEVEL.JAS "R
C****          R          +        LEVEL
      LVVPOS =        JAS
      LVVTYP =   3
      LVFUNC=      LEVEL
      LVVARG=          R
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD             "              R
               R = LV1    AAD
      FAIL=.FALSE.
      IF(STACK(JSTOP,4) .LT. 0) STACK(JSTOP,4)=0
      RETURN
   40 FAIL=.TRUE.
      RETURN
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      GO TO 25001
      END
      SUBROUTINE SQUEEZ
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      INTEGER A,D,BLANK,AICH
      DATA BLANK/1H /,AICH/1HH/
      J=0
      DO 10 I=1,M
      IF(D(I) .EQ. HLANK) GO TO 10
      J=J+1
      D(J)=D(I)
      IF(D(J) .NE. AICH) GO TO 10
      IF(JTYP .NE. 3) GO TO 10
      M=M+J-I
      RETURN
   10 CONTINUE
      M=J
      RETURN
      END
      SUBROUTINE SSTOP(FAIL)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      INTEGER START,STOP,ASSOC,STACK,STR(1),STJ,R,TEMP
      LOGICAL FAIL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      JSTOPS=JSTACK
C   5 R+ASSOC/10
    5 CONTINUE
C****          R             +        ASSOC
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       ASSOC
      LVVARG=            R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1     AAD =            R
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          10
      JSTOPS=JSTOPS+1
      STACK(JSTOPS,1)=R
      STACK(JSTOPS,2)=0
      STACK(JSTOPS,4)=0
C  10 R(=STOP//20,+STOP/30 "R/5/5)
   10 CONTINUE
      LV1     AAD =            R
C**** LV1     AAD            =          STOP
      LVVAL = -100
      IF (LV1     AAD.NE.       STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          20
C**** LV1     AAD            +          STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        STOP
      LVVARG= LV1     AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1     AAF = LV1     AAD
      IF (LVVAL.NE.-1) LV1     AAF = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          30
C**** LV1     AAF            "            R
              R = LV1     AAF
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          5
      IF (LVVTR.NE.-1) GO TO          5
   20 FAIL=.FALSE.
      RETURN
C  30 JSTACK=JSTOPS/40
   30 CONTINUE
C****     JSTACK            =       JSTOPS
      LVVAL = -100
      IF (     JSTACK.NE.     JSTOPS) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          40
      FAIL=.TRUE.
      RETURN
   40 R=STACK(JSTOPS,1)
      JAS=STACK(JSTOPS,2)+1
C     R+ASSOC.JAS "TEMP//50
C****          R            +        ASSOC
      LVVPOS =         JAS
      LVVTYP =    3
      LVFUNC=       ASSOC
      LVVARG=            R
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D) .
      LV1    AAD =           R
      IF (LVVAL.NE.-1) LV1     AAD = LVVAL
C**** LV1    AAD            "        TEMP
            TEMP = LV1    AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          50
      JSTOPS=JSTOPS-1
C     //30
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          30
   50 STACK(JSTOPS,2)=JAS
C     R=TEMP//40
C****          R            =        TEMP
      LVVAL = -100
      IF (          R.NE.       TEMP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          40
C     TEMP "R//5
C****        TEMP           "          R
               R =       TEMP
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          5
C     COMPLETE
      RETURN
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      GO TO 25001
      END
      SUBROUTINE STATNO
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,BLANK,STATRA
      INTEGER BITPUT,BITGET
      DATA BLANK/1H /
      LOC=0
      DO 5 I=1,5
      IF(A(I) .NE. BLANK) GO TO 10
    5 CONTINUE
      IF(ITYP .NE. 18) GO TO 7
      IF(IBLKDT .EQ. 1) RETURN
      IF(NBRNCH .EQ. 0) GO TO 110
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBRNCH,6)
      RETURN
    7 IF(ITYP .EQ. 28) GO TO 90
      IF(NBLOCK .EQ. 0) GO TO 8
      IF(IBLOCK(NBLOCK) .EQ. 998) GO TO 32
      IF(NB .EQ. 2) GO TO 31
      IF(NB .EQ. 1) GO TO 70
      RETURN
    8 NBLOCK=1
      GO TO 34
   10 IF(ITYP .EQ. 18) GO TO 50
      JPTR=I
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 50
      IF(A(6) .NE. BLANK) GO TO 50
      IF(JPTR .LT. 6) GO TO 50
      CALL STSRCH
      IF(BITGET(STATRA(2,LOC),9,3) .EQ. 1) GO TO 60
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,9)
      IF(LTYP .EQ. 9) GO TO 20
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),ITYP,6)
      GO TO 30
   20 STATRA(2,LOC)=BITPUT(STATRA(2,LOC),9,6)
   30 IF(ITYP .EQ. 28) RETURN
      IF(NBLOCK .EQ. 0) GO TO 8
      IF(NB .EQ. 1) GO TO 32
   31 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=998
      NBRNCH=1
   32 NB=0
      NBLOCK=NBLOCK+1
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBLOCK,24)
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBRNCH,6)
      J1=IBLKST+1
      J2=NBLOCK-NBRNCH-1
      IF(IBLOCK(J2) .GT. 6000) GO TO 34
      J21=J2-1
   21 IF(IBLOCK(J1) .LT. 6000) GO TO 34
```

```
      IRES=IBLOCK(J1)
      DO 22 K2=J1,J21
   22 IBLOCK(K2)=IBLOCK(K2+1)
      IBLOCK(J2)=IRES
      GO TO 21
   34 IBLKST=NBLOCK
      NBRNCH=0
      IBLOCK(IBLKST)=BITPUT(0,ILOOP,12)
      IF(LOC .EQ. 0) RETURN
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),LOC,32)
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),IBLKST,32)
      IF(BITGET(STATRA(2,LOC),15,3) .NE. 1) RETURN
      IF(IOVFLW .EQ. 1) RETURN
      IF(LOC .NE. ISTACK(1,ILOOP)) GO TO 80
      IF(ITYP .GE. 3 .AND. ITYP .LE. 6) GO TO 100
      IF( ITYP .EQ. 9 .OR. ITYP .EQ. 10 .OR. ITYP .EQ. 17) GO TO 100
      NB=2
      ISTACK(2,ILOOP)=1
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=6000+ISTACK(4,ILOOP)
      KLOOP=ILOOP-1
      DO 40 J=1,KLOOP
      LOOP=ILOOP-J
      IF(ISTACK(2,LOOP) .EQ. 1) GO TO 40
      IF(ISTACK(1,LOOP) .EQ. LOC) GO TO 35
      ILOOP=LOOP
      RETURN
   35 ISTACK(2,LOOP)=1
      NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=6000+ISTACK(4,LOOP)
   40 CONTINUE
      ILOOP=0
      RETURN
   50 IERC=32
      GO TO 200
   60 IERC=33
      GO TO 200
   70 IERC=34
      GO TO 200
   80 IERC=35
      GO TO 200
   90 IERC=36
      GO TO 200
  100 IERC=37
      GO TO 200
  110 IERC=38
  200 CALL ERROR(IERC,IDM1,IDM2,IDM3,IDM4)
      RETURN
      END
      SUBROUTINE STFNC(NFNC)
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
      INTEGER BITGET
      NARG=BITGET(IDTBL(3,LOC),7,6)
      NAR2=IFNCRA(NFNC,1)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),[illegible]2)
      NARGS=MIN0(NARG,NAR2)
      KOUNT=0
      NT=1+(NARG-1)/3
      DO 10 I=1,NT
      ICOL1=-6
      ICOL2=-3
      DO 10 J=1,3
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARGS) RETURN
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      IF(BITGET(IFNCRA(NFNC,I+1),ICOL2,3) .NE. 0)
     $ CALL ERROR(50,KOUNT,I1,I2,I3)
      ITP1=BITGET(IFNCRA(NFNC,I+1),ICOL1,3)
      IF(ITP1 .EQ. 0) GO TO 10
      ITP2=BITGET(IDTBL(3,LOC+KOUNT),10,3)
      IF(ITP1 .NE. ITP2) CALL ERROR(51,KOUNT,I1,I2,I3)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE STORE
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      NID=NID+1
      IF(NID .GT. 500) GO TO 20
      IF(INITID(IDTYP) .NE. 0) GO TO 5
      INITID(IDTYP)=NID
    5 CONTINUE
      IDTBL(1,NID)=NXTID(1)
      IDTBL(2,NID)=NXTID(2)
      IDTBL(4,NID)=0
      IF(LASTID(IDTYP) .EQ. 0) GO TO 10
      LAST=LASTID(IDTYP)
      IDTBL(4,LAST)=NID
   10 LASTID(IDTYP)=NID
      RETURN
   20 WRITE(6,25)
   25 FORMAT(/////5X,46H SYMBOL TABLE OVERFLOW - PROCESSING TERMINATED)
      STOP
      END
```

```
      SUBROUTINE STSRCH
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      INTEGER STATRA
      IF(NLABEL .EQ. 0) GO TO 15
      DO 10 I=1,NLABEL
      IF(STATRA(1,I) .NE. N2) GO TO 10
      LOC=I
      RETURN
   10 CONTINUE
   15 NLABEL=NLABEL+1
      IF(NLABEL .GT. 200) GO TO 20
      LOC=NLABEL
      STATRA(1,LOC)=N2
      RETURN
   20 WRITE(6,25)
   25 FORMAT(/////5X,53H STATEMENT NO. TABLE OVERFLOW - PROCESSING TERMI
     *NATED)
      STOP
      END
      SUBROUTINE SUB
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH(10),IALPH2(8),KT(5)
      INTEGER BLANK,COMMA,RPAR,A,D
      INTEGER BITPUT
      DATA IALPH/1HS,1HU,1HB,1HR,1HO,1HU,1HT,1HI,1HN,1HE/
      DATA IALPH2/1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN/
      DATA KT/1HR,1HC,1HD,1HI,1HL/
      DATA BLANK/1H /,LPAR/1H(/,RPAR/1H)/,COMMA/1H,/
      DATA IF/1HF/
      ISTATE=0
      NARG=0
      IF(ITYP .EQ. 30) GO TO 5
    2 DO 3 I=1,8
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50
    3 CONTINUE
      GO TO 12
    5 DO 10 I=1,10
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 50
   10 CONTINUE
      GO TO 17
   12 IPTR=JPTR
      IST=1
      IFRST=NEXT(IST)
      IF(IFRST .EQ. IF) GO TO 14
      DO 13 I=1,5
      IF(IFRST .NE. KT(I)) GO TO 13
      ISTATE=I
      GO TO 14
   13 CONTINUE
   14 JPTR=IPTR
   17 CALL GNLE
      IF(JTYP .NE. 2) GO TO 50
      IDTYP=2
      CALL STORE
      IE=D(1)
      IF(ITYP .NE. 31) GO TO 15
      IF(NEXT(JPTR) .NE. LPAR) GO TO 50
      IFNCNM=NID
      GO TO 20
   15 IF(NEXT(JPTR) .EQ. BLANK) GO TO 30
      IF(A(JPTR-1) .NE. LPAR) GO TO 50
   20 CALL GNLE
      IF(JTYP .NE. 2) GO TO 60
      CALL SEARCH
      IF(ISRCH(1) .NE. 0 .OR. ISRCH(2) .NE. 0)
     $ CALL ERROR(86,NXTID(1),NXTID(2),IDM2,IDM3)
      IDTYP=1
      CALL STORE
      IDTBL(3,NID)=BITPUT(IDTBL(3,NID),1,12)
      NARG=NARG+1
      IF(NEXT(JPTR) .EQ. RPAR) GO TO 30
      IF(A(JPTR-1) .NE. COMMA) GO TO 50
      GO TO 20
   30 LOC=1
      D(1)=IE
      IF(NARG .GT. 63) CALL ERROR(83,IDM1,IDM2,IDM3,IDM4)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NARG,7)
      IF(ITYP .EQ. 30) RETURN
      IF(ISTATE .EQ. 0) GO TO 55
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)
      RETURN
   55 CALL IMPTYP
      RETURN
   50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
      RETURN
   60 CALL ERROR(86,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
      END
      SUBROUTINE SUBCHK
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2)-JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     $LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      INTEGER BITPUT,BITGET
      NSUBS=NSUBS+1
      IF(NSUBS .GT. 100) GO TO 50
      NARG=BITGET(IDTBL(3,1),7,6)
      ITP=BITGET(IDTBL(3,1),10,3)
      IF(BITGET(IDTBL(3,1),18,1) .EQ. 1) GO TO 15
      IDTBL(3,1)=BITPUT(IDTBL(3,1),1,18)
      DO 5 I=1,NLIST
      IF(IDTBL(1,1) .NE. ISUBLT(1,I)) GO TO 5
      IF(IDTBL(2,1) .NE. ISUBLT(2,I)) GO TO 5
      LISTLC=I
      IDTBL(3,1)=BITPUT(IDTBL(3,1),LISTLC,32)
      GO TO 20
    5 CONTINUE
      CALL ERROR(52,IDM1,IDM2,IDM3,IDM4)
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,1)
      ISUBLT(2,NLIST)=IDTBL(2,1)
      ISUBS(NSUBS)=NLIST
      ISUBLT(3,NLIST)=0
      ISUBLT(4,NLIST)=0
      IDTBL(3,1)=BITPUT(IDTBL(3,1),NLIST,32)
      IF(NARG .EQ. 0) RETURN
      IPTR=NINTFC+1
      ISUBLT(3,NLIST)=BITPUT(0,NARG,6)
      ISUBLT(3,NLIST)=BITPUT(ISUBLT(3,NLIST),ITP,13)
      ISUBLT(4,NLIST)=IPTR
      NINTFC=IPTR+(NARG-1)/3
      KOUNT=0
      DO 10 I=IPTR,NINTFC
      INTFAC(I)=0
      ICOL1=-6
      ICOL2=-3
      DO 10 J=1,3
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARG) RETURN
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      ITP=BITGET(IDTBL(3,KOUNT+1),10,3)
      NDIM=BITGET(IDTBL(3,KOUNT+1),7,6)
      IDTBL(3,KOUNT+1)=BITPUT(IDTBL(3,KOUNT+1),1,15)
      INTFAC(I)=BITPUT(INTFAC(I),ITP,ICOL1)
   10 INTFAC(I)=BITPUT(INTFAC(I),NDIM,ICOL2)
      RETURN
   15 LISTLC=BITGET(IDTBL(3,1),32,9)
   20 ISUBS(NSUBS)=LISTLC
      KLAS=BITGET(ISUBLT(3,LISTLC),10,4)
      NAR2=BITGET(ISUBLT(3,LISTLC),6,6)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,1),IDTBL(2,1),IDM3,IDM4)
      NARGS=MIN0(NARG,NAR2)
      IF(ITP .NE. BITGET(ISUBLT(3,LISTLC),13,3))
     $ CALL ERROR(49,IDTBL(1,1),IDTBL(2,1),IDM3,IDM4)
      IF(NARGS .EQ. 0) RETURN
      IPTR=ISUBLT(4,LISTLC)
      NDPTR=IPTR+(NARGS-1)/3
      KOUNT=0
      DO 25 I=IPTR,NDPTR
      ICOL1=-6
      ICOL2=-3
      DO 25 J=1,3
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARGS) RETURN
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      ITP=BITGET(INTFAC(I),ICOL1,3)
      NDIM=BITGET(INTFAC(I),ICOL2,3)
      ITP2=BITGET(IDTBL(3,KOUNT+1),10,3)
      NDIM2=BITGET(IDTBL(3,KOUNT+1),7,6)
      IOSTAT=BITGET(INTFAC(I),ICOL2+2,2)
      IF(IOSTAT .EQ. 2 .OR. KLAS .EQ. 0) IOSTAT=1
      IDTBL(3,KOUNT+1)=BITPUT(IDTBL(3,KOUNT+1),IOSTAT,15)
      IF(NDIM .NE. NDIM2) CALL ERROR(50,KOUNT,IDM2,IDM3,IDM4)
      IF(ITP2 .NE. 0) GO TO 23
      ITP2=1
      IFST=BITGET(IDTBL(1,KOUNT+1),8,8)
      IF(IFST .LE. 213 .AND. IFST .GE. 201) ITP2=4
   23 IF(ITP .NE. ITP2) CALL ERROR(51,KOUNT,IDM2,IDM3,IDM4)
   25 CONTINUE
      RETURN
   50 CALL ERROR(25,IDM1,IDM2,IDM3,IDM4)
      STOP
      END
```

```
      SUBROUTINE SWITCH
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DO 20 I=1,NID
      IF(IDTBL(4,I) .NE. LOC) GO TO 20
      IDTBL(4,I)=IDTBL(4,LOC)
      IF(LASTID(1) .EQ. LOC) LASTID(1)=I
      GO TO 30
   20 CONTINUE
      INITID(1)=IDTBL(4,LOC)
   30 LAST=LASTID(2)
      IDTBL(4,LAST)=LOC
      IDTBL(4,LOC)=0
      LASTID(2)=LOC
      CALL ERROR(87,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
      RETURN
      END
      SUBROUTINE SYMTAB
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LABELS/STATRA(2,200),NLABEL
      COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
      COMMON/STFUNC/NSTFNC,ISTFNC(10)
      DIMENSION ITABEL(9),IRA(2)
      INTEGER TYPE(2,5),DIMS(3),BITGET,STATRA
      DATA TYPE/4HREAL,1H ,4HCOMP,3HLEX,4HDOUB,
     * 2HLE,4HINTE,3HGER,4HLOGI,3HCAL/
      DATA IBLANK/1H /,IRA(1)/4HARRA/,IRA(2)/1HY/,IFP/4HF.P./,
     * ISLASH/2H///
      DATA DIMS/1H1,1H2,1H3/
      IF(NID .LE. 1) RETURN
      IF(IBLKDT .EQ. 1) GO TO 2
      INTL=INITID(2)
      WRITE(6,1) IDTBL(1,INTL),IDTBL(2,INTL)
    1 FORMAT(//////45X,25H SYMBOL TABLE FOR MODULE ,A4,A2)
      GO TO 4
    2 WRITE(6,3)
    3 FORMAT(//////46X,28H SYMBOL TABLE FOR BLOCK DATA)
    4 LOC=INITID(1)
      IF(LOC .EQ. 0) GO TO 28
      WRITE(6,5)
    5 FORMAT(//56X,9HVARIABLES/30X,4HNAME,12X,4HTYPE,31X,10HRELOCATION)
  100 ITABEL(1)=IDTBL(1,LOC)
      ITABEL(2)=IDTBL(2,LOC)
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 0) GO TO 27
      I=BITGET(IDTBL(3,LOC),10,3)
      ITABEL(3)=TYPE(1,I)
      ITABEL(4)=TYPE(2,I)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 16
      ITABEL(5)=IRA(1)
      ITABEL(6)=IRA(2)
      I=BITGET(IDTBL(3,LOC),7,6)
      ITABEL(7)=DIMS(I)
      GO TO 18
   16 ITABEL(5)=IBLANK
      ITABEL(6)=IBLANK
      ITABEL(7)=IBLANK
   18 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 0) GO TO 20
      ITABEL(8)=IFP
      ITABEL(9)=IBLANK
      GO TO 25
   20 IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1) GO TO 22
      ITABEL(8)=IBLANK
      ITABEL(9)=IBLANK
      GO TO 25
   22 ICOMNM=IDTBL(9,LOC)
      ITABEL(8)=IDTBL(1,ICOMNM)
      ITABEL(9)=IDTBL(2,ICOMNM)
      IF(ITABEL(8) .EQ. IBLANK) ITABEL(8)=ISLASH
   25 WRITE(6,26) (ITABEL(I),I=1,9)
   26 FORMAT(30X,A4,A2,11X,A4,A3,14X,A4,A1,1X,A1,7X,A4,A2)
```

```
27 LOC=IDTBL(4,LOC)
   IF(LOC .NE. 0) GO TO 100
28 IF(IBLKDT .EQ. 1) GO TO 60
   LOC=IDTBL(4,INTL)
   IF(LOC .EQ. 0) GO TO 60
   WRITE(6,31)
31 FORMAT(//55X,10H EXTERNALS/44X,4HNAME,10X,4HTYPE,10X,4HARGS)
30 ITABEL(1)=IDTBL(1,LOC)
   ITABEL(2)=IDTBL(2,LOC)
   LISTLC=BITGET(IDTBL(3,LOC),32,9)
   IF(LISTLC .EQ. 0) GO TO 39
   ITP=BITGET(ISUBLT(3,LISTLC),13,3)
   IF(ITP .EQ. 0) GO TO 32
   ITABEL(3)=TYPE(1,ITP)
   ITABEL(4)=TYPE(2,ITP)
   GO TO 35
32 ITABEL(3)=IBLANK
   ITABEL(4)=IBLANK
35 IF(BITGET(ISUBLT(3,LISTLC),14,1) .EQ. 1) GO TO 37
   ITABEL(5)=BITGET(ISUBLT(3,LISTLC),6,6)
   WRITE(6,36) (ITABEL(I),I=1,5)
36 FORMAT(44X,A4,A2,8X,A4,A3,8X,I2)
   GO TO 39
37 WRITE(6,38) (ITABEL(I),I=1,4)
38 FORMAT(44X,A4,A2,8X,A4,A3,8X,2H>1)
39 LOC=IDTBL(4,LOC)
   IF(LOC .NE. 0) GO TO 30
60 IF(NSTFNC .EQ. 0) GO TO 40
   WRITE(6,62)
62 FORMAT(//50X,20H STATEMENT FUNCTIONS/
  $ 44X,4HNAME,10X,4HTYPE,10X,4HARGS)
   DO 70 I=1,NSTFNC
   LC=ISTFNC(I)
   ITP=BITGET(IDTBL(3,LC),10,3)
   NRG=BITGET(IDTBL(3,LC),7,6)
70 WRITE(6,36) IDTBL(1,LC),IDTBL(2,LC),TYPE(1,ITP),TYPE(2,ITP),NRG
40 IF(NLABEL .EQ. 0) GO TO 50
   WRITE(6,42)
42 FORMAT(//51X,17H STATEMENT LABELS)
   WRITE(6,45) (STATRA(1,I),I=1,NLABEL)
45 FORMAT(40X,5I8)
   DO 47 I=1,NLABEL
   IF(BITGET(STATRA(2,I),9,3) .NE. 1)
  $ CALL ERROR(15,STATRA(1,I),IDM2,IDM3,IDM4)
   IF(BITGET(STATRA(2,I),12,3) .NE. 1)
  $ CALL ERROR(16,STATRA(1,I),IDM2,IDM3,IDM4)
47 CONTINUE
50 LOC=INITID(3)
   IF(LOC .EQ. 0) RETURN
   WRITE(6,52)
52 FORMAT(//53X,14H COMMON BLOCKS/50X,4HNAME,10X,6HLENGTH)
51 ITABEL(1)=IDTBL(1,LOC)
   IF(ITABEL(1) .EQ. IBLANK) ITABEL(1)=ISLASH
   ITABEL(2)=IDTBL(2,LOC)
   WRITE(6,55) ITABEL(1),ITABEL(2),IDTBL(5,LOC)
55 FORMAT(50X,A4,A2,8X,I8)
   LOC=IDTBL(4,LOC)
   IF(LOC .NE. 0) GO TO 51
   RETURN
   END
```

```
      SUBROUTINE TYPE
      COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
     * NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
     * LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      DIMENSION IALPH1(7),IALPH2(7),IALPH3(4),IALPH4(7),IALPH5(15),
     1 IDIM(3)
      INTEGER A,RPAR,COMMA,BLANK
      INTEGER BITPUT,BITGET,COMLOC
      DATA IALPH1/1HL,1HO,1HG,1HI,1HC,1HA,1HL/
      DATA IALPH2/1HI,1HN,1HT,1HE,1HG,1HE,1HR/
      DATA IALPH3/1HR,1HE,1HA,1HL/
      DATA IALPH4/1HC,1HO,1HM,1HP,1HL,1HE,1HX/
      DATA IALPH5/1HD,1HO,1HU,1HB,1HL,1HE,1HP,1HR,1HE,1HC,1HI,1HS,1HI,
     * 1HO,1HN/
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /
      MUL=1
      IT=ITYP-18
      GO TO (10,20,30,40,50),IT
   10 DO 15 I=1,7
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 110
   15 CONTINUE
      ISTATE=4
      GO TO 60
   20 DO 25 I=1,4
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 110
   25 CONTINUE
      ISTATE=1
      GO TO 60
   30 DO 35 I=1,15
      IF(NEXT(JPTR) .NE. IALPH5(I)) GO TO 110
   35 CONTINUE
      MUL=2
      ISTATE=3
      GO TO 60
   40 DO 45 I=1,7
      IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 110
   45 CONTINUE
      MUL=2
      ISTATE=2
      GO TO 60
   50 DO 55 I=1,7
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 110
   55 CONTINUE
      ISTATE=5
   60 ISUB=0
      INCR=MUL
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 110
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 62
      IDTYP=1
      CALL STORE
      LOC=NID
   62 IF(BITGET(IDTBL(3,LOC),11,1) .NE. 0) GO TO 120
```

```
      IF(NEXT(JPTR) .NE. LPAR) GO TO 87
      ISUB=1
      IE=LOC
      I=0
   68 I=I+1
      CALL GNLE
      IF(JTYP .NE. 5) GO TO 65
      IDIM(I)=N2
      IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
      INCR=INCR*N2
      GO TO 67
   65 IF(JTYP .NE. 2) GO TO 110
      CALL SEARCH
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
      IF(ISRCH(1) .EQ. 1) GO TO 66
      IDTYP=1
      CALL STORE
      LOC=NID
   66 IF(BITGET(IDTBL(3,LOC),12,1) .NE. 1)
     $ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 130
      CALL IMPTYP
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
     $ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,13)
      IDIM(I)=2**17+LOC
   67 IF(NEXT(JPTR) .EQ. COMMA) GO TO 68
      IF(A(JPTR-1) .NE. RPAR) GO TO 110
      K=NEXT(JPTR)
      LOC=IE
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)
     $ CALL ERROR(11,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)
      IF(I .GT. 3) GO TO 110
      DO 80 J=1,I
   80 IDTBL(4+J,LOC)=IDIM(J)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)
   87 IF(INCR .EQ. 1) GO TO 90
      IF(BITGET(IDTBL(3,LOC),16,1) .NE. 1) GO TO 90
      COMLOC=IDTBL(9,LOC)
      IDTBL(5,COMLOC)=IDTBL(5,COMLOC)+INCR-1
   90 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)
      IF(A(JPTR-1) .EQ. COMMA) GO TO 60
      IF(NEXT(JPTR) .EQ. BLANK) RETURN
  110 CALL ERROR(7,IDM1,IDM3,IDM2,IDM4)
      RETURN
  120 CALL ERROR(12,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
  130 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
      RETURN
      END
```

Auxiliary Programs and Associated Data

Program GRAPH

```
      DIMENSION INT(1000)
   10 FORMAT(12I6)
   20 FORMAT(2I6.A4,2X,I6,A4,2X,7I6/A4,2X,2I6,A4,2X,3I6,A4,2X,I6,A4,2X,
     $2I6/A4,2X,+I6,A4,2X,3I6,A4,2X,2I6/A4,2X,2I6,A4,2X,8I6/A4,2X,I6,A4,
     $2X,I6,A4,2X,I6,A4,2X,3I6,A4,2X,A4,2X/3I6,A4,2X,3I6,A4,2X,3I6,A4,
     $2X/7I6,A4,2X,4I6/5I6,A4,2X,6I6/(12I6))
      READ 10,(INT(I),I=1,11)
      WRITE(4) (INT(I),I=1,11)
      READ 20,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT(1)
      WRITE(4) INT(1)
      READ 10,(INT(I),I=1,34)
      WRITE(4) (INT(I),I=1,34)
      STOP
      END
```

Syntax Graph

```
  1000    99    17     8    17     0    22    95     0     0     1
   359   60+            1-           2   103   103    67     3     3     4
 (          72     5RL         0   669   483,         10CP     10000   349
 *           7   480   795   795DP     20000    21     8*           9    42
 =          18   480)        930    10   935     6   103   935   480   256
 *          12*         235=          13a           14   346    47*     OR
    15   795   795?         16   391   411NT          17   322   359X
    19    28    47   359    29   935   977+        110    67   318   318
    29   228   359    40   935/        130     3    72    11    72    60
    29    60   999    99   100   101   102   103   104   105   106    40
   107    62   109   111   112   113   114   115   116   117   118   119
    10   120   122   123   124   125   126   127    81    62   128    21
   131   133   134   135   136   137   157   138    93   140    95   142
   144   145   146   147    81   480   148   151   152   153   154   155
   156   110   157   159    93   114    95   160   164   165   166   167
   168   285    60   169   172   173    86   174    88   130   176   179
   133   114    72    95   180   185   186   187   188   189   516   190
   192   193   194   195    86   130   196   199   133   200   202   203
   204    95   205   207   208   209   210   211   212   103   126   208
```

```
213    217    208    218    220    221    222    223    224    710    225    553
227    182    229    231    480    292    232    235    126    236    150    238
566    240    242    243    244    157    245    247    248    182    249    251
252    253    254    255    256    257    258    318    150    259    262    175
263    591    265    593    267    269    270    271    272    273    274    228
275    277    278    279    280    281    282    283    284    175    285    287
288    289    290    291    292    293    294    228    295    297    298    299
300    301    302    256    630    303    306    307    378    308    263    310
312    313    314    315    316    317    208    480    318    321    322    346
323    325    326    327    328    329    263    330    332    333    334    288
335    337    338    339    340    341    342    343    670    235    344    347
348    349    350    351    352    353    354    288    355    357    358    418
359    361    315    362    364    365    366    367    368    322    369    371
372    285    373    375    376    377    378    379    380    381    315    382
384    711    385    387    388    322    389    391    392      3    393    285
395    397    398    399    353    400    292    402    404    405    406    407
408    409    410    411    412    346    413    415    416    417    418    419
353    420    422    423    424    425    426    753    318    427    430    431
385    432     86    434    436    437    438    439    440    441    442    443
444    445    446    447    896    448    450    451    385    452    454    445
346    455    110    458    460    461    462    463    464    465    378    466
468    359    796    469    472    473    474    475    476    477    411    478
480    133    481    483    484    485    486    487    378    488    490    491
445    492    494    495    555    496    498    499    500    501    502    503
504    505    506    507    508    509    510    511    445    512    840    514
516    517    518    519    520    521    522    523    524    525    519    526
418    528    483    530    532    533    487    534    536    537    538    539
452    540    542     28    543    545    546    547    548    549    483    550
552    553    487    445    554    557    558    885    559    561    452    562
564    565    566    567    568    569    570    571    572    573    574     60
529    575    578    579    580    581    582    583    584    585    586     72
587    589    480    590    592    483    593    595    529    596    208    598
600    601    602     88    603    931    605    519    607    609    610    611
612    613    614    615    942    616    618    619    620    621    622    623
624    625    626    627    628    629    630    631    632    633    634    635
636    527    637    639    640    641    642    643    644    645    646    647
648    649    650    651    978    652    654    655    982    656    658    659
660    661    662    663    664    665    666    667    668    669    322    670
672    673    674    675    676    677    678    679    680    681      8    682
684     11    685    687    688    689    690    603    691    693    694    695
696    697    698    699    700    701    702    703    704    705    706    707
708    157    709    711    712    603    713    715    669    716    718    719
720    721    722    723    724    725    726    727    728    729    730    731
732    733    734    735    669    736    738    739    740    741    742    743
744    745    746    747    748    749    750    751    752    753    754    755
756    757    758    759    760    761    762    763    764    765    766    767
768    769    770    771    772    773    774    775    776    777    778    779
780    781    108    782    784    785    786    787    788    789    790    791
792    445    793    795    796    797    710    752    452    798    802    803
804    805    806    807    808    809    810    811    812    813    814    815
816    817    818    752    819    821    822    823    824    235    825    827
828    829    830    483    831    318    833    835    836    837    838    839
840    841    842    843    844    845    846    847    848    839    849    851
852    853    854    855    856    857    858    859    860    861    862    863
864    865    866    867    868    869    870    871    872    873    874    875
876    877    878    879    880    881    882    883    884    885    886    887
888    889    890    891    892    893    894    895    896    897    898    899
900    901    902    903    904    795    905    907    908    519    909    911
912    913    914    915    916    917    918    919    920    921    922    923
924    884    925    927    928    929    930    931    932    933    934    935
936    937    938    939    930    940    942    943    944    935    945    947
948    839    949    951    952    953    954    955    956    957    958    959
960    961    962    963    964    965    966    967    968     28    969    971
972    973    974    975    976    889    977    979    980    981    982    983
896    984    986    987    988    989    990    991    992    993    994    995
996    997    998    889
 43     89      4    683      6     66      8    149    710    657     21     24
```

```
    12    89   617    17   268    19   800    44    28    23   783    13
    26   228   130    29   820    31   241    89   266    35   305    37
    38    36   130    15     4   345   129   191    89     4   130   378
    50   428    94    89    54   471    56   515   445    67    42    61
   560   710   839    65   606     5   346    69   653    12   896    73
   686    15     4    78    93   535    10    33   889    15    84   555
    86   577    89    87    88    10     8    89   597   386    93    98
   519   108   100   101   102   103   104   105   106   107   109    96
   111    93   112   113   114   115   116   117   118   119   120   122
     4   123   124   125   126   127   128   131     1   175   133     4
   134   135   136   137   138   140   182   142   133   144   452   145
   146   147   148   151     7   130   152   153   154   155   156   157
   159   208   160   164   157   228    89   165   166   167   168   169
   172   315     5   173   174   176    89   179    89    29   180   185
   208   555    11    89   186   187   188   189   190   192    20   193
   194   195   196   199    10   603   200   202   235   203   204   205
   207    16   208   209   210   211   212   213   217     5    89   110
   218   220   452   221   222   223   224   225   227   483   229    25
   231   133   232   235   130   235   236   238     3   240   157   242
    30   243   244   245   247    89   248   249   251   150   252   253
   254   255   256   257   258   259   262    89     3   263   265   519
   267    80   269    16   270   271   272   273   274   275   277   256
   278   279   280   281   282   283   284   285   287     3   288   289
   290   291   292   293   294   295   297   285   298   299   300   301
   302   303   306   346    34   307   308   310   411   312    29   313
   314   315   316   317   318   321     1   130   322   323   325   445
   326   327   328   329   330   332   292   333   334   335   337   263
   338   339   340   341   342   343   344   347    59     2   348   349
   350   351   352   353   354   355   357   889   358   359   361    89
   362   364   256   365   366   367   368   369   371   208   372   373
   375    89   376   377   378   379   380   381   382   384   285   385
   387    51   388   389   391   126   392   393   395    21   397    13
   398   399   400   402   208   404     1   405   406   407   408   409
   410   411   412   413   415    89   416   417   418   419   420   422
    89   423   424   425   426   427   430    49     5   431   432   434
    29   436   529   437   438   439   440   441   442   443   444   445
   446   447   448   450   935   451   452   454   418   455   458    95
    13   460   288   461   462   463   464   465   466   468    89   469
   472     4    53   473   474   475   476   477   478   480   378   481
   483   263   484   485   486   487   488   490    13   491   492   494
    29   495   496   498   603   499   500   501   502   503   504   505
   506   507   508   509   510   511   512   514   480   516    55   517
   518   519   520   521   522   523   524   525   526   528    62   530
     3   532   669   533   534   536    76   537   538   539   540   542
    89   543   545    47   546   547   548   549   550   552   710   553
   554   557    83     6   558   559   561    60   562   564     5   565
   566   567   568   569   570   571   572   573   574   575   578    81
    85   579   580   581   582   583   584   585   586   587   589   529
   590   592     7   593   595    14   596   598    95   600   110   601
   602   603   605   669   607    64   609    89   610   611   612   613
   614   615   616   618    40   619   620   621   622   623   624   625
   626   627   628   629   630   631   632   633   634   635   636   637
   639     5   640   641   642   643   644   645   646   647   648   649
   650   651   652   654    68   655   656   658    90   659   660   661
   662   663   664   665   666   667   668   669   670   672   385   673
   674   675   676   677   678   679   680   681   682   684     3   685
   687    72   688   689   690   691   693    89   694   695   696   697
   698   699   700   701   702   703   704   705   706   707   708   709
   711   182   712   713   715     9   716   718   208   719   720   721
   722   723   724   725   726   727   728   729   730   731   732   733
   734   735   736   738    88   739   740   741   742   743   744   745
   746   747   748   749   750   751   752   753   754   755   756   757
   758   759   760   761   762   763   764   765   766   767   768   769
   770   771   772   773   774   775   776   777   778   779   780   781
   782   784    22   785   786   787   788   789   790   791   792   793
   795   114   796   797   798   802    89    18   487   803   804   805
   806   807   808   809   810   811   812   813   814   815   816   817
   818   819   821    28   822   823   824   825   827   322   828   829
   830   831   833   752   835    40   836   837   838   839   840   841
   842   843   844   845   846   847   848   849   851   884   852   853
   854   855   856   857   858   859   860   861   862   863   864   865
   866   867   868   869   870   871   872   873   874   875   876   877
   878   879   880   881   882   883   884   885   886   887   888   889
   890   891   892   893   894   895   896   897   898   899   900   901
   902   903   904   905   907    15   908   909   911    62   912   913
   914   915   916   917   918   919   920   921   922   923   924   925
   927   930   928   929   930   931   932   933   934   935   936   937
   938   939   940   942   977   943   944   945   947     3   948   949
   951     4   952   953   954   955   956   957   958   959   960   961
   962   963   964   965   966   967   968   969   971   710   972   973
   974   975   976   977   979    89   980   981   982   983   984   986
    89   987   988   989   990   991   992   993   994   995   996   997
   998   999    99     3
    43     2     4     3     6     5     8     7     9    90    11    13
    12    14    40    17    16    19    18    44    21    23    22    24
    26    25   110    29    28    31    30    32    80    35    34   717
    38    37   191    15    41    59     1    20    45    46   276    52
    50    49    94   324    54    53    56    55   394    58    42    61
    60   737   311    65    64    66   459    69    68   433   370    73
    72   139   158    78    77    76   493    33    82   178    84    83
    86    85    89    88    87    10   241    92    95    51    93    98
    97    96     0     0     0     0     0     0     0     0     0   108
     0    27     0     0     0     0     0     0     0     0     0     0
   121     0     0     0     0     0     0     0   129   130     0   132
     0     0     0     0     0     0    74     0   141     0   143     0
     0     0     0     0   149   150     0     0     0     0     0     0
     0    75     0     0   161   162   163     0     0     0     0     0
     0   170   171     0     0     0   175     0   177    81     0     0
   181   182   183   184     0     0     0     0     0     0    39     0
     0     0     0     0   197   198     0     0   201     0     0     0
     0   206     0     0     0     0     0     0     0   214   215   216
     0     0   219     0     0     0     0     0     0   226     0   228
     0   230     0     0   233   234     0     0   237     0   239     0
    91     0     0     0     0   246     0     0     0   250     0     0
     0     0     0     0     0     0     0   260   261     0     0   264
     0   266     0   268     0     0     0     0     0     0     0    47
     0     0     0     0     0     0     0     0     0   286     0     0
     0     0     0     0     0     0     0   296     0     0     0     0
     0     0     0   304   305     0     0     0   309     0    63     0
     0     0     0     0     0     0   319   320     0     0     0    48
     0     0     0     0     0     0   331     0     0     0     0   336
     0     0     0     0     0     0     0     0   345   346     0     0
     0     0     0     0     0     0     0   356     0     0     0   360
     0     0   363     0     0     0     0     0     0    71     0     0
     0   374     0     0     0     0     0     0     0     0   383     0
     0   386     0     0     0   390     0     0     0    57     0   396
     0     0     0     0   401     0   403     0     0     0     0     0
     0     0     0     0     0   414     0     0     0     0     0     0
   421     0     0     0     0     0     0   428   429     0     0     0
    70     0   435     0     0     0     0     0     0     0     0     0
     0     0     0     0   449     0     0     0   453     0     0   456
   457     0    67     0     0     0     0     0     0     0   467     0
     0   470   471     0     0     0     0     0     0     0   479     0
     0   482     0     0     0     0     0     0   489     0     0     0
    79     0     0     0   497     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0   513     0   515     0
     0     0     0     0     0     0     0     0     0     0   527     0
   529     0   531     0     0     0   535     0     0     0     0     0
   541     0     0   544     0     0     0     0     0     0   551     0
     0     0   555   556     0     0     0   560     0     0   563     0
     0     0     0     0     0     0     0     0     0     0     0   576
   577     0     0     0     0     0     0     0     0     0     0   588
     0     0   591     0     0   594     0     0   597     0   599     0
     0     0     0   604     0   606     0   608     0     0     0     0
     0     0     0     0   617     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0   638     0     0     0     0     0     0     0     0     0     0
     0     0     0     0   653     0     0     0   657     0     0     0
     0     0     0     0     0     0     0     0     0     0   671     0
     0     0     0     0     0     0     0     0     0     0   683     0
     0   686     0     0     0     0     0   692     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0   710     0     0     0   714     0     0    36     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0    62     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0   783     0     0     0     0     0     0     0     0     0
     0   794     0     0     0     0   799   800   801     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0   820     0     0     0     0     0   826     0     0
     0     0     0   832     0   834     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0   850     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0   906     0     0     0   910     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0   926     0     0     0     0     0     0     0     0     0     0
     0     0     0     0   941     0     0     0     0   946     0     0
     0   950     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0   970     0     0
     0     0     0     0     0   978     0     0     0     0     0     0
   985     0     0     0     0     0     0     0     0     0     0     0
     0     0     0  1000
    96    76    99    97    99    97    96    96    76    97    76    97
    99    76    97    99    97    96    96    99    76    99    97   236
    99    97    72    96    96    99    97    76    97    99    97   232
    99    97    72    99    77    97    96    97    76    77    72    72
    99    97    99    72    99    97    99    97    72    76    99    99
    97    72    72    99    97   236    72    99    97    73    72    99
    97    73    73    96    76    96    72    99    72    73    96    96
    96    96    96   236    96    99    73    76    96    97    96    96
    76    96     0     0     0     0     0     0     0     0     0   236
     0    76     0     0     0     0     0     0     0     0     0     0
    77     0     0     0     0     0     0     0   236    76     0    77
     0     0     0     0     0     0    76     0    76     0    76     0
     0     0     0     0   236    76     0     0     0     0     0     0
     0    76     0     0    76    76    76     0     0     0     0     0
     0    76    77     0     0     0    76     0    76    76     0     0
    76    76    77    76     0     0     0     0     0     0   236     0
     0     0     0     0    77    76     0     0    76     0     0     0
     0    77     0     0     0     0     0     0     0    77    76    76
     0     0    76     0     0     0     0     0     0    76     0   236
     0    76     0     0    76    76     0     0    77     0    76     0
   236     0     0     0     0    76     0     0     0    76     0     0
     0     0     0     0     0     0     0    76    77     0     0    76
     0   236     0   236     0     0     0     0     0     0     0    76
     0     0     0     0     0     0     0     0     0    77     0     0
     0     0     0     0     0     0     0    76     0     0     0     0
     0     0     0    76   236     0     0     0    76     0    76     0
     0     0     0     0     0     0    77    76     0     0     0    76
     0     0     0     0     0     0    76     0     0     0     0    76
     0     0     0     0     0     0     0     0   236    77     0     0
     0     0     0     0     0     0     0    76     0     0     0    76
     0     0    76     0     0     0     0     0     0    76     0     0
     0    76     0     0     0     0     0     0     0     0    76     0
     0   236     0     0     0    76     0     0     0    76     0    77
     0     0     0     0    76     0    77     0     0     0     0     0
     0     0     0     0     0    76     0     0     0     0     0     0
    76     0     0     0     0     0     0   236    77     0     0     0
    76     0    76     0     0     0     0     0     0     0     0     0
     0     0     0     0    76     0     0     0    76     0     0    76
    77     0    76     0     0     0     0     0     0     0    76     0
     0    77   236     0     0     0     0     0     0     0    76     0
     0    76     0     0     0     0     0     0    77     0     0     0
    76     0     0     0    76     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0    76     0   236     0
     0     0     0     0     0     0     0     0     0     0    76     0
    77     0    76     0     0     0   236     0     0     0     0     0
    76     0     0    76     0     0     0     0     0     0    76     0
     0     0   236    77     0     0     0   236     0     0    77     0
     0     0     0     0     0     0     0     0     0     0     0    76
   236     0     0     0     0     0     0     0     0     0     0    76
     0     0    77     0     0    77     0     0   236     0    76     0
     0     0     0    76     0   236     0    76     0     0     0     0
     0     0     0     0   236     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0    77     0     0     0     0     0     0     0     0     0     0
     0     0     0     0   236     0     0     0   236     0     0     0
     0     0     0     0     0     0     0     0     0     0    76     0
     0     0     0     0     0     0     0     0     0     0   236     0
     0   236     0     0     0     0     0    76     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0    76     0     0     0    77     0     0    76     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0    76     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0   236     0     0     0     0     0     0     0     0     0
     0    76     0     0     0     0    76   236    76     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0   236     0     0     0     0     0    76     0     0
     0     0     0    76     0    76     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0    76     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0    77     0     0     0    76     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0    76     0     0     0     0     0     0     0     0     0     0
     0     0     0     0    76     0     0     0     0    76     0     0
     0    77     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0    76     0     0
     0     0     0     0     0    76     0     0     0     0     0     0
    76     0     0     0     0     0     0     0     0     0     0     0
     0     0     0    77
     0
     8   391   982   349   942   516   553   591   630   670   711   753
   796   840   885   931   978    42    11    48    68    89   111   675
   154   642   137   593   108   566    19    29   228   483
```

Program SESLIST

```
      INTEGER TYPE,CLASS,CLASS2,BLKSIZ(100),BLKTYP(100)
      DIMENSION ISUBLT(4,200),INTFAC(600),IARGS(200),NAME(2)
      EQUIVALENCE (BLKTYP(1),IARGS(1)),(BLKSIZ(1),IARGS(101))
      EQUIVALENCE (ITEMP,RTEMP)
      NLIST=0
      IPTR=1
    1 READ(5,5,END=40) NAME,NARGS,TYPE,CLASS,ISIZE
    5 FORMAT(A4,A2,1X,3I2,1X,I6)
      IF(ISIZE .LT. 1) ISIZE=0
      NLIST=NLIST+1
      IVAR=0
      ISUBLT(1,NLIST)=NAME(1)
      ISUBLT(2,NLIST)=NAME(2)
      IF(NARGS .NE. -1) GO TO 7
      NARGS=1
      IVAR=1
    7 NARG2=NARGS
      CLASS2=CLASS
      CALL SHIFTL(CLASS2,22)
      CALL SHIFTL(NARG2,26)
      CALL SHIFTL(TYPE,19)
      RTEMP=OR(ISIZE,CLASS2)
      RTEMP=OR(RTEMP,NARG2)
      RTEMP=OR(RTEMP,TYPE)
      ISUBLT(3,NLIST)=ITEMP
      ISUBLT(4,NLIST)=0
      IF(NARGS .EQ. 0) GO TO 1
      CALL SHIFTL(IVAR,18)
      RTEMP=OR(ISUBLT(3,NLIST),IVAR)
      ISUBLT(3,NLIST)=ITEMP
      ISUBLT(4,NLIST)=IPTR
      IF(CLASS .EQ. 7) GO TO 25
      JPTR=IPTR
      INC=1+(NARGS-1)/3
      IPTR=IPTR+INC
      NDPTR=IPTR-1
      NPARAM=3*NARGS
      READ(5,10) (IARGS(I),I=1,NPARAM)
   10 FORMAT(20(3I1,1X))
      KOUNT=0
      DO 22 I=JPTR,NDPTR
      ITEMP=0
      NSHIFT=32
      DO 20 K=1,9
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NPARAM) GO TO 22
      NSHIFT=NSHIFT-3
      ISHIFT=NSHIFT
      IF(MOD(KOUNT,3) .EQ. 0) ISHIFT=NSHIFT+1
      CALL SHIFTL(IARGS(KOUNT),ISHIFT)
   20 RTEMP=OR(RTEMP,IARGS(KOUNT))
   22 INTFAC(I)=ITEMP
      GO TO 1
   25 JPTR=IPTR
      IPTR=IPTR+NARGS
      NDPTR=IPTR-1
      READ(5,27) (BLKSIZ(I),BLKTYP(I),I=1,NARGS)
   27 FORMAT(10(I6,1X,I1))
      KOUNT=0
      DO 30 I=JPTR,NDPTR
      KOUNT=KOUNT+1
      CALL SHIFTL(BLKSIZ(KOUNT),15)
      CALL SHIFTL(BLKTYP(KOUNT),12)
      RTEMP=OR(BLKSIZ(KOUNT),BLKTYP(KOUNT))
   30 INTFAC(I)=ITEMP
      GO TO 1
   40 WRITE(4) NLIST,NDPTR
      WRITE(4) ((ISUBLT(I,J),I=1,4),J=1,NLIST)
      WRITE(4) (INTFAC(I),I=1,NDPTR)
      WRITE(6,50) NLIST
   50 FORMAT(//42X,37H NEW LIST HAS BEEN CREATED CONTAINING,I4,6H NAMES)
      STOP
      END
```

Basic Interface Definition File

```
ABS      1 1 4
102
AIMAG    1 1 4
202
AINT     1 1 4
102
ALOG     1 1 4
102
ALOG10   1 1 4
102
AMAX0   -1 1 4
402
AMAX1   -1 1 4
102
AMIN0   -1 1 4
402
AMIN1   -1 1 4
102
AMOD     2 1 4
102 102
ATAN     1 1 4
102
ATAN2    2 1 4
102 102
CABS     1 1 4
202
CCOS     1 2 4
202
CEXP     1 2 4
202
CLOG     1 2 4
202
CMPLX    2 2 4
102 102
CONJG    1 2 4
202
COS      1 1 4
102
CSIN     1 2 4
202
CSQRT    1 2 4
202
DABS     1 3 4
302
DATAN    1 3 4
302
DATAN2   2 3 4
302 302
DBLE     1 3 4
102
DCOS     1 3 4
302
DEXP     1 3 4
302
DIM      2 1 4
102 102
```

```
DLOG     1 3 4
302
DLOG10   1 3 4
302
DMAX1   -1 3 4
302
DMIN1   -1 3 4
302
DMOD     2 3 4
302 302
DSIGN    2 3 4
302 302
DSIN     1 3 4
302
DSQRT    1 3 4
302
EXP      1 1 4
102
FLOAT    1 1 4
402
IABS     1 4 4
402
IDIM     2 4 4
402 402
IDINT    1 4 4
302
IFIX     1 4 4
102
INT      1 4 4
102
ISIGN    2 4 4
402 402
MAX0    -1 4 4
402
MAX1    -1 4 4
102
MIN0    -1 4 4
402
MIN1    -1 4 4
102
MOD      2 4 4
402 402
REAL     1 1 4
202
SIGN     2 1 4
102 102
SIN      1 1 4
102
SNGL     1 1 4
302
TAN      1 1 4
102
TANH     1 1 4
102
```

INITIAL DISTRIBUTION

Copies	
10	NAVSEA PMS304-32 White
2	NAVSEA PMS405-40 Cuthbert
12	DDC

CENTER DISTRIBUTION

1	18/1809
1	1802.2 Frenkiel
1	1802.4 Theilheimer
1	1809.3 D. Harris (Central Depository, CMLD)
1	182 Camara
1	1826 Culpepper
30	1826 Wybraniec
1	184 Lugt
1	185 Corin
1	186 Sulit
1	189 Gray
1	1890 Taylor
30	5214.1 Reports Distribution
1	522

Microfiche copies	
30	1826 Wybraniec

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.